Web-based Modelling with Applications to Surgical Training

By

Ying Li

Submitted in accordance with the requirements for the degree of Doctor of Philosophy.

The University of Leeds
School of Computing

November 2002

The candidate confirms that the work submitted is her own and that the appropriate credit has been given where reference has been made to the work of others.

Abstract

Web-based surgical training has the potential to offer a cheap training platform, accessible from anywhere in the world, at any time. However the use of web-based VR for training faces a number of challenges including fast collision detection and real-time deformable modelling. In this thesis, these problems are addressed and solutions are given to meet these challenges.

In collaboration with the neurosurgeons at Leeds General Infirmary, the use of web-based VR to train neurosurgeons is investigated, using the percutaneous rhizotomy procedure as a case study. A number of surgical tasks are identified. The training requirements that can be met using a web-based approach, exploiting VRML and Java, are described. A novel solution for collision detection is presented. To provide an effective training platform, an assessment tool is developed and integrated in the training simulator to allow the progress of trainees to be monitored.

For the deformable modelling of soft objects, the extension of the existing 3D ChainMail algorithm, which is only able to model a volumetric object represented as a uniform rectilinear mesh, is achieved in two ways. First, the algorithm is modified so that it can be applicable to a surface represented as a non-uniform rectilinear mesh. The modified algorithm, called SurfaceChainMail in this thesis, is implemented in the simulation of the cutting of soft tissue.

Secondly, the 3D ChainMail algorithm is further extended to arbitrary meshes, without sacrificing the computation speed. This new algorithm is named as the Generalised ChainMail algorithm. An important aspect of this work is that both surfaces and volumes can be handled by the same basic approach: a surface represented as a triangular mesh in 3D space is deformed using the same algorithm as a volume represented as a 3D tetrahedral mesh.

Acknowledgements

I would like to acknowledge the following people who have given me support and encouragement during this research:

My supervisor, Professor Ken Brodlie, for his faithfully understanding, support, guidance and encouragement throughout this work.

Nick Phillips, the neurosurgeon from LGI hospital, for his continual support.

To Peter Coltman for his encouragement, suggestions and user feedback for the simulator.

My colleagues for their encouragement and support: Roger Boyle, Andy Bulpitt, Selan Dos Santos, Marcelo Cohen, Nuha El-Khalili, Stuart Lovegrove, Jason Wood, Richard Holbrey, Nuk Puangpronpitagand, Riri Sari, Yasmi Aziz, and Adriano Lopes.

My friends, Haijuan, Hailan, Liyan, and Yao for sharing hard times with me.

My father, who died in June 2001, my mother, and my sisters for their love and continual support.

My husband, Haibo, and our son, Ruolong, for their love, patience, encouragement and acceptance of me. My little daughter, Lucy, who has accompanied me and given me enormous happiness all the time since she was born.

Declarations

Some parts of the work presented in this thesis have been published or submitted in the following articles:

Y. Li, K. Brodlie, N. Phillips. *Soft-tissue modelling for web-based surgical simulation* in: Westwood, J D, Hoffman, H M, Mogel, G T, Robb, R A & Stredney, D (editors) Medicine Meets Virtual Reality 2002, pages 261-267, IOS Press. 2002.

K. Brodlie, N. El-Khalili, Y. Li. *Using Web-based computer graphics to teach surgery*. Computers & Graphics, vol. 24, pages 157-162, 2000.

Y. Li, K. Brodlie, N. Phillips. *Web-based VR training simulator for percutaneous rhizotomy* in: Westwood, J D, Hoffman, H M, Mogel, G T, Robb, R A & Stredney, D (editors) Medicine Meets Virtual Reality 2000, pages 175-181, IOS Press. 2000.

Y. Li, K. Brodlie. *Soft Object Modelling with Generalized ChainMail - Extending the Boundaries of Web-based Graphics.* Computer Graphics Forum, 2003. (Accepted)

Contents

List of Figures

List of Tables

Chapter 1 Introduction

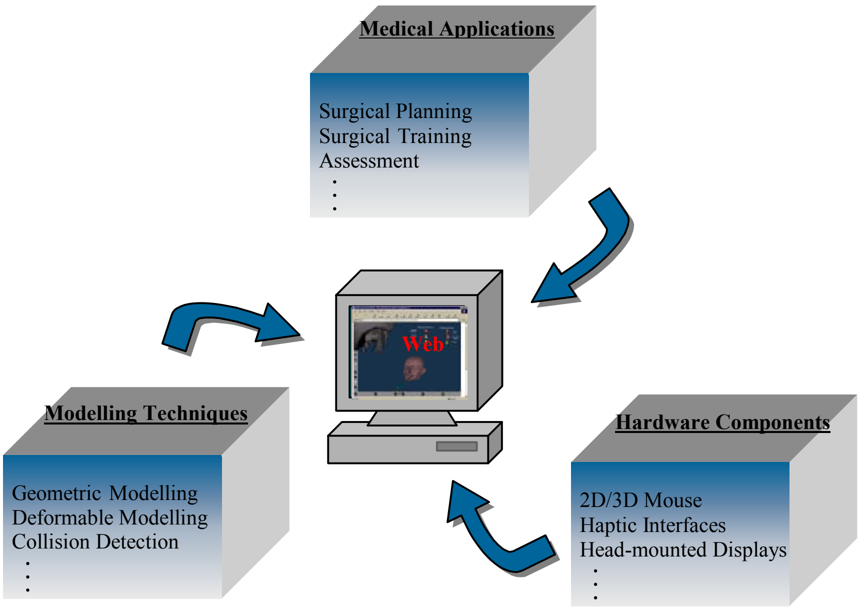


Figure 1.1: A virtual reality system for medical applications

1.1 Motivation

Virtual reality (VR) is having an impact on many aspects of medicine. Indeed, advanced modelling techniques and hardware are merging together to create an increased believable sense of reality and therefore producing useful tools for diverse disciplines in medicine. One of the beneficial areas is surgical training. Current surgical training methods include watching a video of the operation, practicing on a model e.g. a plastic model, practicing on an animal or a cadaver, and undertaking a real operation on a patient with supervision from an experienced surgeon in order to acquire skills needed in a surgical procedure. This situation exposes the problems of expense, demand of resources, and risks to patients. Moreover, there are an increased number and complexity of new procedures, which require new skills to perform. The traditional training methods no longer meet the needs of these emerging techniques. For example, Minimum Invasive Surgery (MIS) is introduced so that trauma to external tissue is minimized. An eye/hand coordination skill is needed in order to operate on the patient from a distance. This skill is difficult to acquire from the traditional training modes. In the meantime, the methods of assessing the trainees' performance are mainly subjective and it is sometimes difficult to distinguish the quality of the performance between different levels of surgeons. As a result, alternative or improved training methods and assessment tools are developed. Experimental evidence (VR Presence, 2002) shows that VR-based surgical training tools are able to deliver efficient and cost-effective training environments to allow trainees to develop their skills, and to facilitate an accurate objective measurement of performance. It has therefore the potential as an additional and improved training and assessment tool.

While some simulators have been demonstrated as high performance trainers for certain applications, there are still factors which limit their use. One of the factors is the cost. Many VR systems still require a high-end workstation to obtain high fidelity results. Although some systems are targeted at PCs, many of them need dedicated VR equipment, such as a force feedback device. Another problem is the limited computation capacity of today's computer, which requires a trade off between realism and speed. Real time simulation not only requires real time rendering – the image displayed on the screen, but also real time interaction – the behaviour displayed on the screen during interaction, e.g. collision detection and deformation. The realistic effects need not only realistic medical models, but also realistic behaviour caused by interaction between the medical models and instruments. During recent years there have been significant advances in this area, many of these make use of the high accuracy in behaviour by using Finite Element Method (FEM) (Bro-Nielsen, 1998). While much improvement in efficiency has been achieved, significant computational resources are still required to achieve a certain level of realism. Moreover, the efficiency gains are often achieved through a pre-processing step dependent on the topology of the meshes, and so the simulation of operations, such as cutting, are not possible in real-time. Furthermore, medical image data, which are generated from real patients or ordinary people, mostly have a complex structure and a large size. In order to achieve high accuracy in interaction with the medical data, it requires simplifying the complex structure to reduce the size of datasets to a certain threshold to get an acceptable frame rate. These may be useful for the applications where the realism of the interaction is crucial (for example, surgical planning), or the psychomotor skill is essential.

However, a realistic model is very useful for surgical training, especially for training the novice medical students to acquire a knowledge of 3D anatomical structure and 3D relations between the instruments and the medical model. Recent work by Gibson (Gibson, 1997) has developed a fast algorithm, 3D ChainMail, which can achieve real time interaction with large datasets so that it can keep the highly realistic features of the medical models. However a major limitation is the restriction to uniform rectilinear grids. Many objects are more conveniently described in terms of arbitrary meshes, and indeed most work using FEM and mass-spring techniques uses tetrahedral meshes. This restriction limits its widespread use, which has been pointed out in several papers (Nienhuys et al., 2001; Bielser et al., 2000).

The emergence of the World Wide Web (WWW) as a powerful distributed computing environment opens up new possibilities for surgical training. In contrast to dedicated VR installations, the delivery platform is a basic PC with browser software. This is extremely attractive as a means of providing training to large groups, at any time and at any place. The restriction to simple compute performance motivates even further an interest in fast algorithms for real-time interaction with medical datasets.

As an important component of today's WWW technology, VRML - the Virtual Reality Modeling Language - offers the ability to transfer 3D worlds across the Internet. A number of useful training tools have been built using this Web technology (see for example, WebSET, 2002), but typically these are limited by the functionality of VRML. For example, collision detection between objects is not supported in VRML, and we need to explore ways of overcoming this restriction in order to support medical applications where collision between instruments and anatomy is fundamental.

Moreover, because the computational resources available on the client-side are not great, simulations have been restricted to rigid objects. The sophisticated FEM simulations required for soft body deformations are not practical in this type of environment. In cases where deformations are critical, a joint client-server solution has been proposed, with the deformable modelling and collision detection deployed on the server. This was attempted for example by El-Khalili et al. (2000), but it proved to be difficult to synchronise the two parts over the network.

1.2 Research Objectives

As mentioned earlier, virtual reality technology has been realized as a solution to the surgical training and objective assessment of surgeon's performance. The need for cost effectiveness and the features that Web technology provides direct this approach to a particular perspective – using the Web to deliver surgical training so that the training software can be accessed at any time, at any place, by anyone with a standard PC and Web browser. There are good tools for Web-based 3D graphics based on ISO standards (VRML, X3D – Web3D, 2002) – VRML browsers are available free for all major platforms. Moreover it is possible to link a simulation process, written in Java, to the VRML environment through a well-defined interface, the External Authoring Interface (EAI).

Therefore, the objective of this study is to investigate the feasibility of delivering useful surgical simulators on the Web. We also see as important an ability to provide feedback on the use of the simulator – both to the trainee as a means of their learning, and to the trainer as a means of assessment. As the performance of

PCs increases, so we can now be more ambitious in the range of simulations that can be delivered in this way – and in particular we can approach the challenge of deformable modelling entirely on the client side. This opens the way in particular to simulations involving deformation of soft tissue. FEM modelling is still beyond the horizon, but there are other approaches, less demanding computationally, that are now within reach. Our requirements will be biased towards real-time interaction, including an ability to cut, as opposed to high accuracy.

1.3 Contributions of the work

The initial contribution of the work was the development of the Web-based surgical simulator, which uses a novel technique - collision detection, based on task analysis. This incorporates the assessment tool, which is based on the analysis of key skills for the surgical procedure.

The further contribution to a fast deformable modelling technique was to generalise the 3D ChainMail idea of Gibson, from its original conception in terms only of objects defined as elements linked as regular, rectangular meshes, to a form where it can be used for surfaces and volumes defined on arbitrary grids. This allows it to be used for a wide range of surgical procedures in which tissue deformation is involved. Indeed it has applications beyond surgical training, to the deformation of any soft objects, and therefore has value in other areas such as facial animation. Two novel approaches were proposed and studied in this work: the SurfaceChainMail and the Generalised ChainMail. A Web-based simulator for a simple cutting procedure using the SurfaceChainMail algorithm has been developed to demonstrate its ability to handle topology changes. The Generalised ChainMail simulation engine has been linked to VRML for display and user interaction, allowing the end-result to be used in Web-based applications.

1.4 Thesis Outline

Chapter 2 reviews the virtual reality technologies that have been applied to the main areas of medical applications including the relevant areas to the work: surgical training, assessment of performance, and web-based training. This chapter also reviews the common deformable modelling techniques including the physically-based and geometric-based.

Chapter 3 describes the development of the PRP (Percutaneous Rhizotomy Procedure) training system. The chapter introduces the medical background of the procedure, followed by the task analysis in order to identify the skills acquired through the functions of the system. The architecture of the Web environment is discussed. Techniques such as object modelling are also described in this chapter. The new collision detection technique for the web-based environment is proposed. Finally, the methodology of interface design including the assessment tool is discussed.

Chapter 4 presents the new technique, SurfaceChainMail - extending the 3D ChainMail algorithm to surface objects and to irregular rectilinear grids. The algorithm and the results are discussed. As an example of the use of this technique for surgical simulation and allowing topology changes, simulation of a cutting procedure is presented.

Chapter 5 describes another new algorithm developed in this work, Generalised ChainMail – extending the 3D ChainMail to surfaces and volumes with arbitrary grids. The action of processing each element no more than once is justified and described in this chapter.

Chapter 6 discusses the evaluations of the PRP surgical training simulator and deformable modelling techniques. The results are analysed and discussed.

Chapter 7 concludes the work described in this thesis. Some suggestions for future work are presented.

Chapter 2 Review

Many researchers believe that virtual reality (VR) technology has great ability (or potential) in supporting a variety of fields in medicine. As a result, a large number of systems have been developed to enhance the performance in reality in these diverse areas. This chapter reviews the systems related to this work.

In the next section, the main areas in medicine where VR technology has been applied are introduced. Some techniques that have been used in the area, surgical training, with which this research is concerned, are described. Section 2.2 reviews Web-based applications of the VR technology in surgical simulation. Some deformable modelling techniques commonly used in surgical training are introduced in section 2.3, which is followed by a discussion and analysis of the limitations of the current technology.

2.1 VR in Medicine

VR technology has mainly been applied to four areas in medicine: diagnosis, therapy, surgery, education and training as shown in Figure 2.1 and described in turn in the following sections.

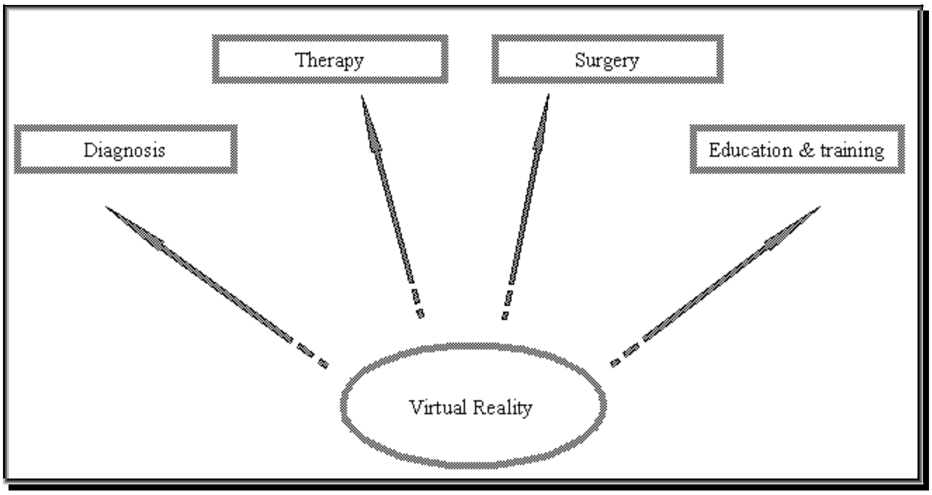


Figure 2.1: VR applications in medicine

2.1.1 Diagnosis

VR is creating new and fascinating ways in assisting diagnosis in medicine. Increasing computer power facilitates the use of visualisation techniques to support three dimensional organ constructions from CT (Computer Tomography) scan or MRI (Magnetic Resonance Imaging) of patient-specific anatomical structures. These anatomical models, along with the conventional medical approaches, provide effective ways to support diagnosis. A typical example is virtual endoscopy (Jolesz et al., 1997; Nain et al., 2001). Endoscopy is a minimally invasive surgery either through natural body orifices or small incisions in a hollow organ with a long, thin instrument. An optical system is fixed at the tip of the instrument to view the inner surfaces of hollow organs. This highly skilled surgery has significant potential benefits to patients, such as reduced pain and recovery time, and it therefore becomes a very important procedure for diagnosis and therapy in various pathologies. However, it lacks the ability of providing information about the anatomy within or beyond the wall of an organ. This problem can be overcome by the use of virtual endoscopy, which uses a 3D patient-specific anatomical model constructed from CT scan or MRI data to provide a better view of the inner structures so that surgeons can explore the model and detect lesions or diseases. The system developed in (Nain et al., 2001) allows the user to interactively explore the internal surface of a 3D anatomical model and to create and update a fly-through trajectory through the model to simulate endoscopy. It is envisaged that the virtual endoscopic procedures could replace real endoscopic investigations in the foreseeable future in some areas of diagnosis (Székely et al., 1999).

To improve the quality of diagnosis of prostate cancer, Zeng et al. (1998) developed a prostate needle biopsy protocol using a computer-based 3D visualisation and simulation system. The prostate models used in the system are constructed from the digitised prostate specimens with localised cancers. A 3D probability map of the prostate cancer spatial distribution is built by incorporating a large number of individual digitised specimens to show the 3D distribution of the prostate cancers. It shows the probability of cancer detection at each location in the prostate. Based on this map, a statistically optimised needle biopsy protocol was introduced by incorporating the needle locations with the highest probability of tumor detection. This system has two simulation modes: an automatic simulation and an interactive simulation. With the automatic simulation, the process of the prostate needle biopsy can be controlled automatically by the computer. On the other hand, with the interactive simulation, six-degree-of-freedom tracking device is integrated to simulate the ultrasound probe used in the actual prostate biopsy procedure. The urologist determines the location for each needle insertion based on a specific protocol under the guidance of the synthesised ultrasound image. Thus, in addition to diagnosis, this simulation system can be used to evaluate the performance of different biopsy protocols. It can also be used as a training or testing system for the residents to practice their skills of biopsy, or a planning system for the urologists before they undergo a complex real biopsy procedure.

The high quality of 3D anatomical images help in diagnosis significantly so that surgeons can have a better view and better knowledge of the organs being examined than reality. This technology and the teleconsultation service enable doctors to verify a diagnosis with a distant partner experiencing the same 3D images without sending the patient to another examination. The system in (Berlage, 1997) was designed for cardiologists and cardiosurgeons. It integrates 3D ultrasound images of the patient with an artificial animated surface model of the heart to provide an implicit but strong communication channel in medical teleconsultation. The scenic communication interface was created to allow partners to exchange information through the virtual worlds by orientation, visual explanation and implicit communication based on PCs with advanced 3D graphics and ISDN communication lines. The orientation is gained by moving the point of view in an animated 3D scene. The visual explanation is achieved through structural models, animation and history of interaction. The implicit communication is performed through the actions during interaction. This system is designed for cardiologists and cardiosurgeons but the concept is transferable to other medical areas.

2.1.2 Therapy

VR based therapy is offering innovative treatment alternatives for patients. Examples include the use of VR for the training and skill enhancement of people with disabilities, for the treatment of eating disorders, phobias, and other psychological problems.

Virtual environments provide people with disabilities safe scenarios to learn and to practise the skills that they often lack such as exploration, navigation, discovery, and moving obstacles – and specific skills such as crossing streets with wheelchairs (Inman et al., 1994; Gunderson et al., 1996). They can also be designed to train patients in certain actions. The research carried out in (Boian et al., 2002) shows promising results from a VR-based system, which was used to rehabilitate post-stroke patients in the chronic phase. The major rehabilitation exercises focus on finger fractionation (finger harmonized movement), finger range of motion, speed and strength to reduce impairments where the patients are equipped with the CyberGlove or the RMII haptic glove to play with different virtual objects for different exercises.

VR has been studied in psychology as a tool for the treatment of different psychological problems: eating disorders (Lozano et al., 2002), and phobias such as fear of public speaking (North et al., 2002; Pertaub et al., 2001). Recent research in (Lozano et al., 2002) has developed a virtual environment system for the treatment of eating disorders. In their system, a virtual kitchen in which there are various virtual foods and a virtual scale is modelled. Patients can interact with the virtual objects to simulate the eating process and compare their own weights with the weights they should have.

In the treatment for phobias, part of the therapeutic process involves the patients being exposed to the phobic stimulus in a safe and controlled environment until the fear itself fades away (Phobia, 2002). Virtual reality is being used to immerse patients in the virtual environment without danger to them (North et al., 1998; North et al., 2002; Schuemie et al., 2002). In virtual exposure, patients utilise immersive devices, such as head mounted display, to control the virtual objects in a phobic environment. An experiment conducted in (Pertaub et al., 2001) placed 40 subjects (e.g. patients) in a virtual seminar with three different virtual audiences (negative, positive and static). This research claimed that people responded to virtual seminar audiences as they would to real audiences. Recent study in (North et al., 2002) has shown that virtual reality therapy is a significant addition to traditional therapies for fear of public speaking as well as other phobias problems.

2.1.3 Surgery

The impact of virtual reality in surgery is being investigated in surgical planning, image-guided surgery (Nakajima et al., 1999; Kettenbach et al., 1997; Nakajima et al., 1997; Jr et al., 2002), remote operation (Bar-Cohen et al., 2001) and training for surgery (Webster et al., 2001; Bro-Nielsen et al., 1999). The VR technologies required in the systems are based on the critical nature or tasks of the simulated surgical procedures.

Surgical Training

Surgeons need extensive training to acquire complex skills before being fully qualified. This is especially true for today's surgeons. The rapid growth of new surgical procedures and reduced physician teaching time results in an increasing need for effective alternatives from which surgeons can gain skills. VR is a promising new medium for surgeons to develop their skills (Tendick et al., 2000). This medium could be either a virtual reality or an augmented reality. Thus many researchers have investigated and developed surgical simulators to meet this need – see examples in (Webster et al., 2001; Bro-Nielsen, 1997b; Bro-Nielsen et al., 1999). A variety of surgical training systems have been introduced and proved potential abilities to provide a training platform on which students can practise and learn skills. The old training approach in surgery, so called "see one, do one and teach one", is being replaced by "see one, practise many, do one and teach one", because virtual reality based training tools allow trainees to practise surgical procedures or tasks as many times as they want before performing a real operation.

Surgical procedures consist of a series of specific tasks. In general, the tasks include dissection and suturing of soft tissues, insertion of instruments, retraction of tissues and so on. Surgical simulation using VR technology is often designed to provide surgeons with a unique way to practise and gain skills. Various VR-based surgical training systems have been developed and designed during the last two decades ranging from a desk-top VR (John et al., 2000) training system to an immersive VR training system (Kaufmann et al., 2000). The skills acquired from the systems for surgical tasks include perceptual motor skills and spatial skills. For surgical procedural training, the skill acquired from the system is to put together a complex procedure from a set of steps (Davis et al., 2002).

Virtual reality technologies provide flexible training tools for users so that they can interact with the tools in different ways. Surgical training systems allow users to interact with the simulated virtual world through input devices such as mouse, keyboard or gloves in either an immersive manner or a non-immersive manner. The challenge is how to make a better choice for medical applications. In some cases, a realistic presence or sense is desirable (Montgomery, 2001; Kaufmann et al., 2000) whereas in other cases, a lower degree of presence or sense is sufficient (El-Khalili et al., 2000). As a result, two categories are identified: immersive surgical training simulation and desk-top based training simulation. This section reviews the surgical training systems based on these categories.

- **Immersive Training Simulation**

By integrating an immersive display with desktop input devices, surgical training can be enhanced in a way that users feel as if he or she is performing a real operation. One example is a pericaridiocentesis trainer (Kaufmann et al., 2000), developed to replace animals and cadavers used in the training of new surgeons in a needle-based procedure. This system uses a phantom haptic interface to control the position and orientation of a virtual cannula. In addition, haptic feedback is utilized to simulate the effect of resistance when the virtual cannula touches and penetrates the skin and chest wall of a patient. Moreover, a flat panel liquid crystal display is used to obtain the stereoscopic images of the virtual environment.

Similarly, in (Webster et al., 2001), a prototype of a haptic simulation system was designed for trainees to practise the basic suturing procedure to improve their skills. In this design, a haptic interface provides force feedback. A 'Reachin Display'™ unit is used to provide the user with visuo-motor alignment and association, and Crystal Eyes are used to provide three-dimensional stereographic images. In addition, a physically based particle model (e.g. a mass-spring model) is used to simulate the soft tissue behaviour during the interaction.

Another immersive simulator was created for long duration space missions in the National Biocomputation Center (Montgomery, 2001) using small, lightweight, low-power computer components for easy-transport on a space mission. The system contains a head-mounted display, a computer system, and a number of tracked surgical instruments with integrated monitoring and information retrieval and a voice input/output subsystem. This surgical training and assistance system allows astronaut-surgeons to maintain their skills, acquire new specialty skills as well as performing surgical planning to minimise overall crew and mission risk.

These effective training tools, however, require high-end hardware, which is too expensive for general use. A number of surgical simulators have therefore been developed using a lower end platform to make the systems more portable and feasible for wide-spread use (Kaufmann et al., 2001; Liu et al., 2001).

- **Desk-top Training Systems**

It is not always necessary to provide immersion in a training simulator. Desk-top based training systems have been intensively studied for a variety of surgical procedures ranging from minimally invasive procedures (Gibson et al., 1997; Kühnapfel et al., 2000; Tendick et al., 2000; Bro-Nielsen, 1997) to traditional surgical procedures (Brown et al., 2001), where realistic visual feedback, force feedback and tactile sensation are the key concerns.

Many minimally invasive simulators are developed for training fundamental skills for a specific application. For example, a system designed by Gibson & Samosky et al. (1997) is to enhance training in arthroscopic knee surgery. The system uses 3D volumetric models generated from patient-specific 3D MR or CT scan images and employed a fast deformable modelling technique, so called 3D ChainMail, to provide visual feedback to the user. A force feed-back device provides the user with a haptic sense. One of the advantages of the system is that the fast deformation algorithm can save the computational resources for the rendering of complex medical data so that the anatomical understanding can be enhanced. Similarly, another training system developed for laparoscopic procedures also provides the user with visual and haptic feedback. The simulated procedure involves inserting a catheter into the cystic duct using a pair of laparoscopic forceps. Visual interactions between 3D virtual organs and the instruments are simulated (Basdogan et al., 2001). The system also includes a pair of force feedback devices interfaced with laparoscopic instruments. This system employs a finite element model to simulate the flexible dynamics of the duct, and a particle model to simulate the catheter. A modal analysis approach is used to reduce the complexity of the dynamic model for the duct to achieve real time interaction. However, this system does not provide tools for assessment of skills.

Some simulators are developed not only for training a range of basic skills but also for the assessment of skills. A virtual environment testbed was developed at San Francisco, Berkeley, and Santa Barbara campuses of the University of California. This testbed integrates tools such as assessing performance of basic perceptual motor skills, training in the use of an angled laparoscope, teaching critical steps of the cholecystectomy and performing a common laparoscopic procedure for training laparoscopic surgical skills (Tendick et al., 2000) into the training platform. A four-DOF haptic interface, a fast collision detection algorithm for detecting the contacts between rigid and deformable objects, parallel processing of physical modelling together with rendering techniques are used to simulate the accurate behaviour of interaction.

VR technology has also been used for traditional surgical procedures (Brown et al., 2001; Montgomery et al., 1999) such as suturing, and cutting. A testbed developed by Bro-Nelson et al. (1998) utilises a variety of techniques such as deformable modelling, computer graphics and force-feedback interfaces for training the removal of a shattered kidney with open surgery. In addition, cutting and bleeding in a range of cases are simulated.

Currently, most surgical training simulators are demo-style systems or under development. However, Bro-Nielsen et al. (1999) have developed a complete commercial ready system, so called PreOp^{TM} Endoscopic Simulator. This simulator integrates multimedia, 3D graphics simulation, and force feedback technology on a PC to train surgeons in performing flexible bronchoscopy procedures. Instead of focusing on motor skills training, this system emphasises procedural training. Before practising the procedure, students can experience pre-procedure guidance, patient data, and videos explaining general aspects of the procedure. After each procedure, they can be provided with a comprehensive evaluation of performance from the system. In addition, the students can feel the appropriate forces responding to their actions with the bronchoscope and the location of the bronchoscope in the virtual bronchi. Moreover, the virtual patient can respond to the user's actions in a physiological manner. However, there is no adequate empirical evidence to show the usefulness of the tool.

In contrast to the above system, another commercially available trainer – MIST^{TM}, was introduced by Virtual Presence Ltd. This is a training simulator for the laparoscopic cholecystectomy procedure. Instead of emphasis on procedural training, it uses abstract geometrical models rather than highly accurate tissue models to allow the trainee to concentrate on the development of key psychomotor skills (VR Presence, 2002). Thus, real time performance can be achieved on low cost PCs.

As noted, there are different levels of sense required for different surgical training systems to simulate various levels of difficulty in surgery. Too much realism may not always be desired (Brodlie et al., 2000)! The challenge is how to apply sufficient realism to each aspect of the application. The degree of realistic presence, geometric representation and interactive behaviour are those factors that need to be balanced. Not only surgical training systems but also surgical planning and image-guided systems need this balance.

Surgical Planning

One of the aims of surgical planning is to find the best approach to surgery preoperatively to minimize damage. Normally, precise 3D anatomical models of patients are used to help surgeons to make decisions. For example, in (Nakajima et al., 1997), a computer-assisted surgical planning system for cerebrovascular neurosurgery was designed for surgeons, using the 3D models reconstructed from MR images, to select appropriate intervention, assess operative risk, find the normal and pathological relationships, select the surgical approach and localize lesions intra-operatively. Accuracy is a critical issue in surgical planning. In the facial surgical simulation presented in (Roth et al., 1998), in order to provide surgeons with accurately predicted 3D models of the post-surgical morphology and appearance of the face before the actual surgery is carried out, finite element methods are applied in the true volumetric soft tissue models combined with accurate geometric models of facial surface and individual skulls to build realistic models.

Image-guided Surgery

The aim of image-guided surgery is to guide surgical procedures intra-operatively with preoperatively acquired 3D models. 3D reconstruction technology can be used to assist anatomical knowledge and surgical orientation (Nakajima et al., 1997) and other functions as well. For example, in (Akatsuka et al., 2000), a navigation system for a surgical microscope and an endoscope was developed for neurosurgery to help the surgeons to manipulate medical instruments within the patient's body. A 3D patient-specific anatomical model of the targeted tumour, as well as other significant anatomical landmarks, were generated and registered with the patient body coordinate frame. During surgery, the 3D model was superimposed onto the live video images taken from the microscope and the endoscope to get navigational information. This system was used in the hospital for clinical tests. The experiments show that the system can be effectively used in actual theatres.

Virtual reality technology can also be used in robot-assisted surgery and remote surgery (Phillips et al., 2000; Bar-Cohen et al., 2001). In (Phillips et al., 2000), a robot-assisted system, KneeCAS-1, was designed to assist accurate pre-operative planning, and improve the implementation and outcome of osteotomy procedures around the knee, using a number of methodologies, such as fitting a generic 3D model of the bone to the available data, then calculating the intersection of the cutting planes with the model to obtain 2D bone contours within the plane. The advantages of remote surgery include reduced presence of surgeons and reduced risk of exposure to infection and disease. For example, if patients are in a critical condition, the surgery can be carried out immediately even if the corresponding surgeon is at a long distance from the patient. Patients with infectious diseases can undergo operations without the presence of surgeons, thus reducing risks to the surgeons.

2.1.4 Education

Virtual reality opens up a new opportunity for surgeons to advance their knowledge and skills by direct or indirect experience through a virtual world. This virtual world allows surgeons to explore and to interact with objects in a safe and controlled way, which would not otherwise be achievable by any other means. Anatomic VisualizeR (Rigamonti et al., 2000) designed a virtual dissection room with a broad range of interactive virtual tools for users to carry out a number of tasks such as displaying, editing, creating cross-sectional views, measuring sizes and distances, identifying structures, and drawing lines and simple objects. In addition, the users can use a searching facility to access additional information resources. Moreover, the data used in the system being gathered from an actual human body, gives more accurate anatomical models compared to the conventional anatomical tools such as plastic models. It would therefore be a better choice for students to acquire anatomical knowledge.

Similarly, VR-based learning tools can be designed to combine the knowledge of anatomy with effective learning activities. In (Dev, 1999), a game was designed to test knowledge of anatomical spatial relationships. Students were required to place a cross-sectional image of a slice through the virtual human body in a correct location, outline and name each of the key structures visible in the section. In addition, an outline of a body cross-section with no visible internal structures was created for each section so that students can place a corresponding slice in the front view again. Moreover, students can keep requesting clues from the system until they feel confident enough to identify the location of the slice. This game proved to be useful for effectively discriminating between people with different levels of exposure to knowledge of anatomy.

2.2 Web-based Applications

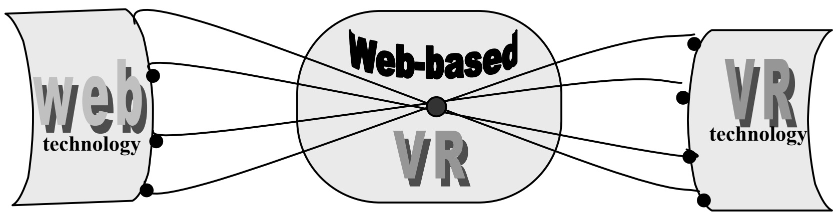


Figure 2.2: A web-based VR system

The great success of the World Wide Web has led to ever-increasing applications in diverse areas for people from different segments of the population. As a result, the impact on medicine is tremendous. The applications include visualisation of medical data from MRI or CT scans on the web (Marovic et al., 1997), medical education on the Web, collaboration between medical staff to share a workspace through a web browser (Chronaki et al., 1997), and web-based surgical simulation (John et al., 2000; El-Khalili et al., 2000).

The advanced Internet technology and virtual reality technology are merging together to create new opportunities for surgeons to enhance their learning and skills in a web-based environment. With these new advances, surgeons are able to pursue training from anywhere, and at any time, via a PC and Web browser. To facilitate a variety of functionality in such innovative systems, a number of researchers have incorporated or extended the new technologies to fulfil the users needs.

A typical Web-based VR application can be decomposed into simulation and presentation processes. The presentation component, handling interaction with the user, necessarily lies on the client-side, but the simulation can be placed either on client- or server-side. As the web is the main driven force, applications can be classified into two categories according to the client/server architecture.

2.2.1 Client-based Applications

In a client-based architecture, the simulation process is entirely implemented on the client side. Once the application and data are downloaded from the web server, the enabling technology such as VRML and Java3D, which is embedded into a web browser, is utilised to display the anatomical 3D models and provide interaction. Examples include the use of VRML for 3D rendering and interaction as described in (John & Riding, 1999; John & Phillips et al., 1999, John et al., 2000; Warrick et al., 1998).

VRML, however, has very restricted features. Many applications need an extension to the VRML browser for a wider range of use. For example, one of the non-supported features is collision detection between objects. John and Riding tackled this problem by using a simple collision detection technique in the lumbar puncture system (John & Riding, 1999). In this system, the instrument path is known in advance and the object to be hit is unique, and so a prediction of a collision occurrence can be made. To achieve real time interaction, some pre-processing, such as the plane equation calculations of all polygons on the model, needs to be performed. This method, however, is only applicable to objects with convex polygons, and only supports object-object collision rather than object-objects collision. To enhance the viewing of high-resolution middle-ear data, Warrick et al. (1998) extended an early VRML browser to support transparent rendering of surfaces. Section images can be superimposed on the model, allowing students to view a section in its 3D context. It allows users to choose the model structures, section images, and associated viewing parameters to costume their desired 3D scene. (Note that later VRML browsers now support transparency).

The major advantage of the client-based architecture is that once the models and applications are downloaded into the client side, users can perform surgical simulation on their own domain without worrying about network bandwidth. However, it has limited flexibility in supporting some functions for some applications. An alternative approach is then needed to fulfil the requirements for those applications as described in the following section.

2.2.2 Server-based Applications

A typical example of a server-based architecture is WebSTer (El-Khalili et al., 2000) – a web-based surgical training simulator for minimally invasive surgery for the treatment of abdominal aortic aneurysms. This system includes three VRML worlds and a Java applet window. It contains an overall viewer, a local viewer, virtual tools and a Java applet window, to display 2D slices through the aorta. A 2D mouse is used to manipulate the virtual tools so as to inserting or rotating the virtual guide-wire or catheter. This is a server-based system in which the jobs are distributed between the server and the client. The major jobs, such as deformable modelling and collision detection, are carried out on the server side. Other jobs, such as the rendering, are done on the client side. These two processes communicate through a socket connection. One of the key advantages of using a server-based approach is that the complexity and performance burden can be kept off the actual client side. Complex computation, which requires a powerful machine to process, can be distributed on the server side, and it therefore reduces the requirements to the client machine. This system is capable of simulating the complex behaviour during interaction, such as deformable modelling and collision detection. However, real time performance is determined by both the computational costs and the speed of network communication.

One feature that a server based architecture can provide is multi-user collaboration. In both (John et al., 2001) and (Mason, 2001) systems, Java-based Deep Matrix software is used as a base platform, which is placed on the server side, for communication between the clients. VRML worlds are utilised to provide visualisation and interaction. Within the shared multi-user group, when one client changes the VRML world, the changes will be made available to the other clients simultaneously. This will allow surgeons to monitor the students' performance and to be able to supervise them from the distance. This is especially useful when web-based training tools are used for training or self-learning. Students need feedback from the experts to guide and correct their performance, and therefore achieve a better performance.

Currently, the WebSET project (John et al., 2001; Riding et al., 2001), carried out in Manchester Visualisation Centre, provides a set of training tools for a variety of surgical procedures. Apart from the collaboration environment, this project is integrating multimedia technologies to support traditional media of text and graphics. In addition, the researchers are trying to provide extra value to the system by incorporating a force-feedback mouse to create haptic effects. This force-feedback mouse is a device with two-degrees of freedom. It is not clear whether this device can help trainees transfer their skills to in vivo. Additionally, this extra device brings extra costs. The question is whether this extra cost adds extra value to the system performance.

2.3 Deformable Modelling

As the entities in virtual environments, the representation of objects plays a crucial role in the simulation systems. The modelling issues associated with the objects originate challenges in a variety of fields. In general, these can be classified into two classes: object modelling and behaviour modelling. Object modelling involves the description of the static structure of objects without any motion, attributes of reaction and topology changes. Behaviour modelling, on the other hand, handles the motion and topological change of objects with attributes of reaction. If the virtual environment only contains rigid objects, the motion, and collision and response of the objects need to be simulated in order to mimic the reality. However, if soft objects are presented in the environment, additional features such as deformable object attributes and topological changing must be included for realistic simulation.

Deformable modelling is based on geometric objects. The objects can be stored and represented in a number of ways. In general, 3D objects fall into two categories: surface representation and volume representation. Surface representation, which defines the surface boundary of an object, can be expressed in different forms including implicit form, explicit form, parametric form and polygonal meshes. Volume representation, which represents both surface boundary and the interior of an object, consists of basic shape primitives such as cube, tetrahedron and prism. One of the main advantages of the surface representation is the reduced number of elements required to define the objects, because no interior elements are involved in the modelling and rendering. However, they do not behave like volumetric 3D objects and lack a defined interior (Bro-Nielsen, 1996), especially when complex inhomogeneous models of the organs are involved and a cutting task is applied to the models. Both approaches have been widely used in various modelling systems. Modelling the deformable behaviour of these objects is a challenging task in the computer graphics community. Many techniques have been demonstrated their feasibility for the applications or potential applications. These applications include cloth simulation (Baraff et al., 1998), and character simulations (James et al., 1999) and so on.

In the medical field, many applications involve the interaction between medical instruments and soft tissue. Thus deformable modelling is essential for simulating the soft tissue behaviour. As a result of the interaction, a set of surgical tasks such as pushing, pulling, cutting and suturing might be integrated into a surgical simulation system. In a scientific analysis and surgery planning system when anatomic features and realistic behaviour in interaction are essential, both realistic geometric models and accurate interaction are needed and real time performance is less important (Nebel, 2001). Surgical training, on the other hand, emphasises on real time behaviour in interaction. This is often achieved by using fast modelling algorithms during interaction (Gibson et al., 1998; Gibson & Samosky et al., 1997), numerical solutions for fast computation (Bro-Nielsen et al., 1996) or by applying a pre-processing step (Bro-Nielsen et al., 1996; Bro-Nielsen, 1998). There are a number of approaches for deformable modelling, varying from pure geometric deformable modelling to physically based modelling. Those methods are based on either surface models or volumetric models. In this section, a review of current research in deformable modelling is carried out, looking at both geometric and physically based approaches.

2.3.1 Geometric Modelling

Geometric modelling involves purely geometric techniques, which are computationally efficient, without simulating physical behaviour. Object deformation is achieved based on spatial algorithms (Sederberg et al., 1986; Coquillart, 1990) or by changing parameters of the function that defines the object (Sinnott et al., 2000) or constraints of the object (Terzopoulos et al., 1991; Borrel et al., 1994).

One example is the work presented in (Borrel et al., 1994). Borrel et al. introduced a Simple Constrained Deformations (Scodef) model to allow users to specify the deformations intuitively. The constraint points on the object are defined arbitrarily. Two elements are associated with each constraint point: the displacement and radius of influence. The displacement determines the distance between the new position and the original position of the constraint point whereas a radius of influence determines how far the deformation will spread out from the constraint point. Each constraint determines a local B-spline basis function, which centralises at the constraint point and drops to zero for points beyond the radius. For each constraint point, the displacement is the greatest. It will be monotonically decreasing until there is no deformation for points beyond the specified radius of the influence region. The deformation at each point on the surface is the combination of the deformations caused by each constraint point. Figure 2.3 shows the deformation at one constraint.

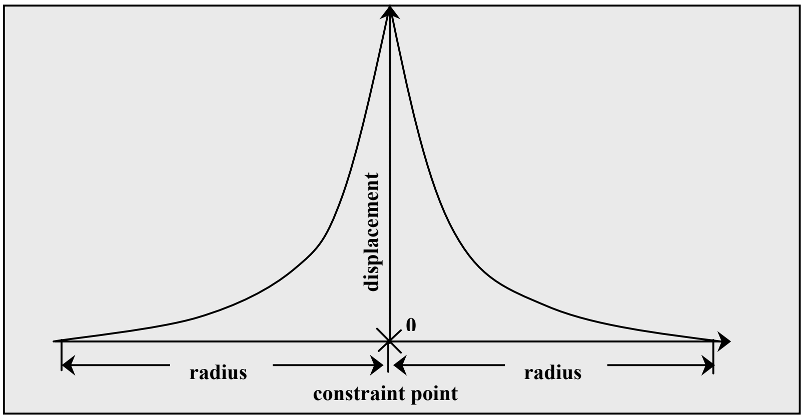


Figure 2.3: The specification of a constraint in the Scodef model

This intuitive deformation technique is very efficient as there is no involvement of physically derived constraints in this model. However, it requires users to specify a displacement, the region of influence for each deformation, which is a tedious job for each object (Thompson, 1995).

Sederberg and Parry (1986) introduced the basic concept of a free-form deformation (FFD), which has been widely used in computer graphics. Free-form deformation is a general geometric method and it can be applied on a solid as well as a surface represented by parametric surfaces patches, polygonal data or other forms locally or globally. The deformed object is embedded into a 3D lattice space. Instead of adjusting individual control points of the object, it is deformed by tuning a higher level of control - 3D lattices. This FFD lattice is a parallelepiped represented by a trivariate Bézier volume. Once this embedding solid is moved by the user through an interface, the changes of the positions of the 3D lattices are passed onto the embedded objects. The deformation of this embedded object is then obtained indirectly. This technique has a number of restrictions, and has therefore been extended by other researchers. Coquillart uses non-parallelepiped 3D lattices to deform an arbitrarily shaped object, either by adding arbitrarily shaped bumps to the object, or by bending it along an arbitrarily shaped curve (Coquillart, 1990; Coquillart et al., 1991). As the parallelepiped lattice restricts arbitrarily shaped deformations, this extension of FFD (EFFD) allows the user to apply different shaped deformations to objects.

As noticed, these FFD and EFFD approaches use indirect control of the deformation through 3D lattices, which makes it difficult to interact with the exact contact points on the surface of the deformed object. This limitation has been overcome by a number of researchers (Hsu et al., 1992; Hu et al., 2001) so that an intuitive interaction can be achieved with the FFD approach.

A further interesting enhancement of the FFD was presented in (Hirota et al., 1999). This approach utilises the physical constraints, namely volume preservation, with geometric deformation of solids. This is achieved by minimising the elastic energy of the geometric deformation and subjecting it to the volume-preserving criterion. The new node positions of the deformation lattice are computed. Large deformations can be handled by this method. Multi-resolution representations of boundary surfaces are adapted to speed up the optimisation convergence. This method can handle any objects, including polyhedra and solids defined by NURBS or Bezier surfaces.

2.3.2 Physically based Modelling

The aim of most surgical simulation is to provide a medical tool as realistically as possible. A realistic tool requires not only realism in the anatomical models, but also realism in the interaction. Non-physical methods for deformation modelling are fast but they do not represent the physical nature of objects being manipulated. On the other hand, physically based modelling methods normally require more computation but they are better able to simulate dynamic behaviour of objects. As computing power and graphics capabilities have increased, there has been a trend towards the use of the physically based methods for deformable modelling. A variety of such methods have been proposed and applied in surgical simulation. In this review five promising approaches are considered, namely mass-spring models, finite element methods, boundary element methods, hybrid methods and 3D ChainMail.

Mass-spring Models

One of the most widely used approaches is mass-spring models (Nedel et al., 1998; Kühnapfel et al., 2000; Brown et al., 2001), where the objects are constructed as a set of nodes with point masses attached to each node and connected by damped springs. The object elements move based on the forces applied to the elements. In a dynamic system, Newton’s Second Law governs the motion of each point:

$$m_i \ddot{x}_i + c_i \dot{x}_i + \sum_j F_{ij} = F_{iext} \qquad (2.1)$$

where, m_i is the mass of the point i, x_i is its position in 3D space. $\ddot{x}_i$ and $\dot{x}_i$ are its acceleration and velocity respectively, c_i is the damping coefficient, $\sum_j F_{ij}$ is the sum of internal forces exerted by its linked mass points, and F_{iext} is the sum of the external forces acted on the mass i. Each spring force, F_{ij}, is determined by the spring stiffness k_i as well as the length of linked springs and the length when the spring compressed or stretched.

Mass-spring models have been successfully used in surgical simulation (Downes et al., 1998; Brown et al., 2001; Kühnapfel et al., 1997; Lee et al., 1995; Keeve et al., 1998; Bro-Nielsen et al., 1998). For example, Brown et al. used a mass-spring model together with constraint-based techniques to simulate the suturing procedure in microsurgery. A fast quasi-static algorithm was adapted to calculate the deformation of the vessels, which assumes that the velocity of user-displaced nodes is small enough and damping large enough to neglect dynamic inertial and damping forces to achieve static equilibrium at each instant. Kuhnapfel et al. (Kühnapfel et al., 1997) have implemented mass-spring models in their KISMET simulation system to simulate realistic mechanical and physical behaviour of the soft tissue for endoscopic surgery of the abdomen. A number of interaction modules, such as grabbing, cutting, clipping and coagulation of virtual tissue and organs, have been simulated. Instead of using static models, which ignore the damping term in the equation (2.1), this system exploits dynamic mass-spring models and NURBS representation of geometrical models together with kinematics handling to simulate the surgery procedures. An application of mass-spring models to facial tissue modelling (an application which requires greater accuracy than most) is presented in (Lee et al., 1995). A facial model consists of five layers in accordance with the structure of real skin, each of which is connected by different springs. Because of the incompressible nature of real human skin and the impenetrable skull of a human head beneath the muscle layer during a facial expression, this system incorporates volume preservation forces, skull penetration forces and nodal restoration forces along with muscle forces into the mass-spring model. Thus, in addition to the spring force term, $\sum_j F_{ij}$, in the equation (2.1), the forces acting on the node are included in the equation to achieve accurate biomechanical modelling. The solution to the second order equations in the system is approximated by using an explicit Euler method.

Interactive simulation using mass-spring systems is feasible with today's desktop systems for many applications, and it is relatively easy to implement, as it requires no continuous parameterisation and matrix inversion. It is realistic for small deformations. Moreover, it facilitates topological changes such as suturing and cutting with reasonable speed. However, it suffers a number of limitations. The discrete model is a significant approximation of the true physics that occurs in a continuous body, and the spring constants are difficult to derive from measured material properties (Gibson & Mirtich, 1997). Another problem is that it produces unrealistic results when large deformations are involved. In addition, for a dynamic mass-spring system, there is a critical stiffness K_c above which the numerical resolution of the system is divergent (Delingette, 1998; Provot, 1995).

Suppose that the system containing n nodes has a given time step size Δt and a given mass, m_{total}/n. The relation between critical stiffness and the time step is given by:

$$\Delta t \approx \frac{1}{\pi}\sqrt{\frac{m_{total}}{nK_C}} \qquad (2.2)$$

This relation indicates that if the stiffness of the model is increased, the time step has to be decreased. Consequently, the cost of computation time of the system will be increased. Otherwise, a large time step would cause numerical instability.

Compared to geometric deformable modelling, the mass-spring model produces superior results in terms of realism (Lee et al., 1995) as it is a physically based approach. Although it has acceptable performance in many applications, it is restricted to the limited size of data. Consequently, when a large dataset is involved in the interaction, an acceptable interactive rate is accomplished by utilizing an alternative fast algorithm, such as 3D ChainMail which will be introduced later in this section.

Finite Element Methods

By contrast, the Finite Element Method (FEM) approach produces more accurate results and more reliable behaviour than the mass-spring approach. The FEM models the object as a continuum, which is approximated by subdividing the object into a number of simple basic elements such as prisms. Each element consists of a number of nodes linked together to define the element's shape. In addition to the choice of its elements, a finite element model is defined by its shape function and its global parameterisation between parameter space Ω and $\Re^3$ ($\Omega \subset \Re^2$ for surfaces and $\Omega \subset \Re^3$ for volumes) (Delingette, 1998). The FEM can model the dynamic behaviour of the continuum as well as static deformation. The internal force can be defined using linear strain and also non-linear strain.

When external forces are exerted on the surface of the continuum, the initial equilibrium is broken and the material of the object absorbs the energy from the external forces acting on it and stores it inside the body. This process causes a change in the shape and therefore results in deformation. A new equilibrium of the deformed shape is reached when the total potential energy of the system is minimum. The equilibrium equation can be expressed by:

$$\mathbf{P} = \mathbf{S} - \mathbf{L}, \qquad (2.3)$$

where **P** is the potential energy**,** **S** is the internal energy, the strain energy, of the deformable object, and **L** is the external energy from the external forces acting on the deformable object. By minimizing **P** with respect to the material displacement over the object, the equilibrium equation is obtained and the deformed shape can be found by solving the equilibrium equation. With a static system, a linear equation is obtained in the form:

$$\mathbf{F} = \mathbf{KU} \qquad (2.4)$$

where **F** is a vector of applied forces, **K** is the stiffness matrix, and **U** is the displacement vector. **K** is numerically integrated over the element volume. Once all the element stiffness matrices and force vectors have been derived, the resulting linear systems for all of the elements are combined into a linear system of the same form. By applying boundary conditions to the linear system, the unknown nodal displacements can be obtained (Gibson & Mirtich, 1997).

By taking into consideration inertia forces and damping forces, a dynamic system has the form:

$$\mathbf{F} = \mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} \qquad (2.5)$$

where **F** is the composite vector of applied forces, **M**, **D** and **K** are the assembled matrices of element mass, damping, and stiffness respectively, and **U** is the composite displacement vector. The mass matrix is defined in a similar way to the stiffness matrix for a single element by numerical integration over the volume of the element. The damping matrix **D** is often constructed as a linear combination of the **M** and **K** matrices (Gibson & Mirtich, 1997). The resulting displacement of the element nodes is obtained by solving this dynamic system using different integration schemes, such as a semi-implicit or explicit scheme.

FEM has attracted much research in computer graphics as it promises advantages in its accuracy (Keeve et al., 1998; Koch et al., 1999; Chen et al., 1998; Zachow et al., 2000). For example, Azar et al. (2000) used non-linear FEM for MR image-guided needle breast procedures to predict a tumor's position during the procedure. The FEM model used 8-node trilinear isoparametric elements for the breast tissue and 3-node triangular isoparametric elements to model the skin. Koch et al. (1996) applied non-linear FEM based on triangular polynomial shape functions to model the facial surface for prediction of facial deformations after craniofacial and maxillofacial surgery.

However, because of the highly computational nature of the method, it is often difficult to achieve real time interaction with today's computer power. The physical behaviour of a soft tissue model may be considered as linear elasticity if the displacements applied to it remain small (Cotin et al., 2000). To meet real time simulation requirements, this linear elastic model is chosen in many applications (Bro-Nielsen, 1996; Cotin et al., 2000). If the object topology remains unchanged during the simulation, the stiffness matrix **K** is constant. This makes it possible to use offline pre-processing, such as the inversion of the stiffness matrix **K**, thus increasing the speed of the interaction significantly.

This scheme has been used in many applications (Bro-Nielsen, 1996; Bro-Nielsen, 1997a; Cotin et al., 1996). Another well-known technique to improve the speed of interaction was introduced by Bro-Neilsen et al. (Bro-Nielsen et al., 1996). They use condensation techniques, where the linear matrix system is compressed by considering only the surface model in computation in a pre-processing. In addition, the Selective Matrix Vector Multiplication scheme is used to simplify the matrix vector multiplication with an assumption that forces are applied on a small area of the object. However, pre-calculation of the stiffness matrix restricts its change during interaction. A topology change will require re-calculating the stiffness matrix during interaction and this is too expensive to process for interactive simulation. Instead of using pre-calculation of the global stiffness matrix, recent research by Nienhuys et al. (Nienhuys et al., 2000; Nienhuys et al., 2001) uses iterative processing techniques to avoid the updating of matrix inverses, and to compute the displacement for each node by gradually moving to a configuration with minimal elastic energy. This static linear system uses tetrahedral elements with linear interpolation function on each tetrahedron, which allows interactive cutting. Modal analysis is another method to speed up the computation. In this method, the mass, damping and stiffness matrices in Equation (2.5) are diagonalised by solving the eigen problem so that the equations of motion (2.5) can be uncoupled (Chen et al., 1998). The solution of the eigen problem is computationally expensive, however, this is a pre-processing stage and it is possible to dramatically reduce the complexity of the FEM model without a great loss in accuracy (Chapman et al., 1997). Chen et al. (1998) implemented two methods to solve the dynamic elastic FEM in their muscle animation system: direct integration and modal analysis, and applied local deformation optimisation with modal analysis for higher resolution muscle volumetric animation.

A linear elastic system can produce realistic results for objects with small displacements. However, it will result in ill-conditioning for large displacements. Zhuang et al. (1999) solved this problem by using nonlinear strain in the FEM model so that large global deformation can be modelled realistically. To achieve real-time performance, they applied a diagonalisation method to decouple the mass matrix into a diagonal matrix so that the nonlinear system of equation (2.5) becomes a set of independent algebraic equations. This diagonalisation is equivalent to converting the distributed mass to a particle system of concentrated mass (Zhuang et al., 2000) so that no matrix inversion is needed. This idea has been adapted in (Wu et al., 2001) for their simulation system.

The most important feature of the FEM is its high degree of accuracy. This feature is most influenced by the properties of the object. Therefore a number of techniques have developed for measuring tissue properties accurately (Fung, 1993; Brouwer et al., 2001). The work carried out in (Brouwer et al., 2001) measures both in vivo and ex vivo properties for different organs. Their results will be available to other researchers to access over the Internet.

So far, FEM has been widely used in surgical planning, offline analysis and predicting the outcomes of surgery. Many researchers are trying to tackle its problems in achieving real time performance as mentioned above so that it can be used for surgical training. At present, real time performance is still problematic, especially for large medical data with complex structures. Therefore, other fast deformable modelling techniques are preferable in complex interactive simulation.

Boundary Element Methods

Another promising approach for deformable modelling is the boundary element method (BEM) (James et al., 1999; Monserrat & Juan et al., 1999; Monserrat et al., 2000). BEM has been used widely in engineering analysis and has been recently used in computer graphics. Compared to FEM, BEM only discretises the surface regions of an object and therefore the system to be solved is smaller and hence computational costs are less. Furthermore, it allows for local modifications of the mesh to be performed in an easier and hence faster way (Monserrat et al., 2000) and hence the simulation of incisions in real time with the BEM is possible. However, the BEM produces a dense system and when the number of elements increases, the computational costs increase dramatically. Monsterrat et al. have tested the BEM with a non-parallelised algorithm and shown that real time behaviour is possible for objects discretised into up to 150 surface nodes (Monserrat et al., 2000). Current systems presented in (James et al., 1999; Monserrat et al., 2000) were applied with linear static models and therefore small displacement is allowed. In addition, the modelled objects are restricted to be homogeneous and isotropic.

Hybrid Approaches

As mentioned earlier, a discrete elastic system becomes problematic when the rigidity of the modelled objects becomes larger. This problem can be solved with a hybrid model, as in (Terzopoulos et al., 1988). Terzopoulos et al. introduced the hybrid model with explicit deformable and rigid characteristics. The deformed objects are decomposed into two components: a reference component and a displacement component. The reference component describes the rigid motion of an undeformed body according to the laws of rigid-body dynamics, whereas the displacement component describes the non-rigid motion based on a linear elastic dynamic model. The deformable body is represented as the sum of these two components. The position (q) of mass elements in the deformed body relative to the body frame is represented in the form:

$$\mathbf{q}(u, t) = \mathbf{r}(u, t) + \mathbf{e}(u, t)$$

where u are the material coordinates of points in a body Ω, t is time, **r**(u, t) is the reference component and **e**(u, t) is the deformation component.

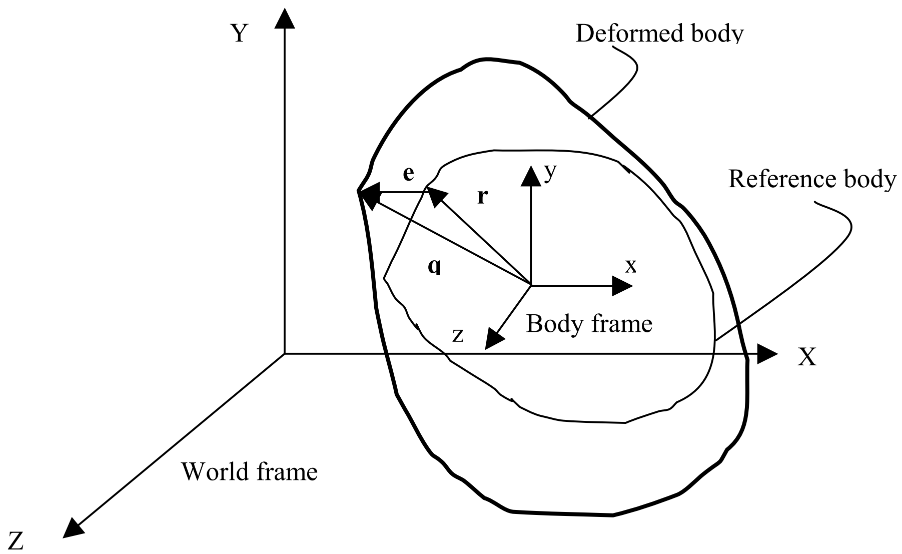


Figure 2.4: A hybrid model

This hybrid elastic approach, as shown in Fig.2.4, has been used in a number of applications (Turner et al., 1993; El-Khalili, 1999). To reduce the computation time, El-Khalili (1999) uses a 1D model for the modelling, but a 3D model for the virtual representation of the virtual tool used in the surgical training system in a web-based environment. Because of the complexity of the elastic model, and the overheads caused by the server-based architecture, real time interaction is difficult to achieve even with very small number of nodes in the system.

As noticed, pre-computation of matrix inversion for a quasi-static linear system based on FEM models can increase speed significantly. However, it is restricted by a fixed topology for real time simulation. A dynamic linear system without pre-processing, on the other hand, requires more computation time but allows topology change. Another hybrid method was therefore introduced in (Cotin et al., 1999; Cotin et al., 2000) to combine those two methods inside a surgical simulator. This is achieved by first specifying the region where topology changes is expected. Then the dynamic linear elastic model is applied on it and the quasi-static model is applied on the rest of the object.

3D ChainMail

Despite attempts to speed up the interaction rate to achieve real time and physically realistic performance, many methods still restrict the number of the elements of the object being modelled. However, most anatomical structures have a rather complex geometry and cannot be realistically represented with such a limited number of elements (Cotin et al., 1999). The 3D ChainMail (Gibson, 1997; Gibson, 1999; Gibson et al., 1998; Schill et al., 1998) algorithm was therefore proposed to tackle the speed problem of deformable modelling. In the ChainMail algorithm, the deformation of an object is determined by two processes. In the first process, a deformation step, one element is moved to a new position, causing each of its neighbours, and their neighbours, and so on, to move to new positions, in a chain reaction. The extent of the movement of an element is governed by geometric constraints which determine the softness of an object. The initial positions of this deformation are then adjusted by a second process, a relaxation step. The positions of the elements are refined so that the system reaches a minimum energy state. The algorithm supports real-time deformation of an object with as many as 125,000 elements.

In the ChainMail process, the object movement is propagated from the user-specified point outwards to its neighbouring elements. This is much like the Scodef model as described earlier, the propagation is carried out from the constraint point outwards to its neighbouring elements. The difference is that 3D ChainMail uses relatively simpler operations on each of the points on the object, thus the computational time is less than the Scodef model. In addition, the number of deformations affects the size of the matrices involved in the operation of the Scodef model, and thus increases the computational costs. In contrast, 3D ChainMail algorithm is only affected by the number of the points involved in the deformation process. As only simple operations are performed on each point in the deformation, the effect to the performance is not significantly large. Moreover, the Scodef model is a pure spatial algorithm without any physical process involved. The 3D ChainMail algorithm, on the other hand, is physically based in the sense that the propagation process is in the form of a chain reaction and also the output of the system energy approaches the minimum. Different behaviour of object material, such as plastic, elastic or rigid, can be modelled with this method. Unlike the mass-spring model, there is no numerical instability problem when the stiffness is large and the acceptable interactive rate can be obtained with a large dataset. In this study of deformable techniques, we are therefore concerned with this approach.

2.4 Discussion

As an alternative environment to the real world, the basic requirements of a VR system are easy to identify: maximum degree of realism and accurate models; real time interaction; easy to use; minimum costs and maximum immersion and presence. However, like other approaches, these requirements are restricted by other factors or by technology. For example, realistic and accurate models, and real time interaction, are limited by current modelling technology and computer power as well as by users' affordability. The degree of immersion and presence is restricted by the technology in hardware design and also the users' affordability. In fact, it is not always necessary to provide a perfect world - perhaps lower achievement in some aspects is sufficient for certain applications. The challenge is how to provide an optimally configured system that balances all these user requirements. This needs to identify what the essential requirement is.

Deciding whether to use immersive VR for the application depends on many circumstances. Although the application itself may be the driving force, the limitations of current technology in hardware and software make it difficult to achieve high levels of usability. This together with additional factors such as costs and portability may lead to the use of lower level desktop VR.

In general, a high degree of realism and accurate anatomical modelling of human organs are of the essence for applications such as surgical planning and image-guided systems. In these applications, efforts of researchers have been directed to providing a maximum degree of realism and accurate anatomical models to help surgeons to achieve highly reliable surgery, and real time interaction is considered less important. On the other hand, surgical training systems demand real time interaction, visual realism of the models of organs and realistic behaviour in interactions, rather than a high degree of accuracy, a reduced accuracy is often acceptable for real time interactivity. How to make the trade off between realism and real time in rendering and interaction is a challenging task.

A web-based environment provides many advantages for people in a variety of disciplines. Surgical training with the web-based environment could give additional benefits to the trainees so that the training program is not restricted to time and the place. They can access the tool at anytime and anywhere with only Internet connection and some plug-ins. These advantages could be revealed with the training tool only when the web technology and VR technology are incorporated appropriately.

Therefore, the initial objective of this research is to develop a web-based training tool for surgical applications. The Percutaneous Rhizotomy Procedure is simulated as a case study. The objects used in this simulation are rigid, which may not always be the case in surgical applications. The research is therefore taken further to investigate the use of the 3D ChainMail algorithm for the modelling of deformable objects in real time. As the algorithm is only applicable to a volumetric object represented as a regular, rectilinear mesh, it will be extended to handle arbitrary meshes to allow much greater flexibility in the arrangement of object elements.

Chapter 3 A Surgical Training System

3.1 Introduction

Surgeons need intensive training to achieve the goal of the acquisition of clinical skills. This is especially true for neurosurgeons. In the UK, a qualified neurosurgeon needs two stages of training: basic surgical training (BST) and higher surgical training (HST). A BST program needs over two years after internship, whereas an HST program needs six years training. This includes the mandatory minimum five clinical years and one flexible year (Joint Committee on Higher Surgical Training, 1999). In addition to this standard training, neurosurgeons need ongoing study throughout their career to update their skills to keep step with rapid advances in neurosurgical techniques.

Neurosurgery encompasses the surgical treatment of the nervous system and its coverings, where the neurosurgeons operate on the brain, spinal cord, the skull and scalp, and the spine (UCI, 2002). The extreme complexity of the nervous system and advanced techniques incorporated into the surgery mean that it is very important to be able to assess a neurosurgeon's ability. There is a need for tools to measure their performance in coordination with the new advances in neurosurgery.

One of the commonly performed procedures in neurosurgery is the percutaneous rhizotomy procedure for the treatment of intractable facial pain. This is a minimally invasive surgery and has become commonly used since the 1960s. This procedure creates a precise lesion of the trigeminal nerve and most patients, even older and frail patients, are able to tolerate the procedure. Moreover, it offers effective treatment with shorter hospitalisation, and earlier return to work and therefore becomes very attractive to the surgeon and patients. This safe procedure might seem to carry minimal risks but this is not always the case. There have been technical hazards with the method and consequently it needs experienced hands.

In open surgery, the anatomical structures of the operational areas are exposed to the surgeons explicitly, which makes it easy for them to make decisions and to perform surgical tasks. The percutaneous rhizotomy procedure involves needle placement and insertion into the target from outside the area of the operation (*Percutaneous* means 'through the skin'.). Although an image-guided system can help the surgeon to perform the procedure, the surgeon still requires extensive spatial skills to relate instruments to anatomy, and procedural skills to make critical decisions. In reality, this is learnt by watching videos and real operations, and by verbal instruction from experienced surgeons. A further step is by participating in a real operation on patients under supervision, which is dangerous. Hence, we need to find new training methods to transfer the skills needed in this procedure to reality, effectively and safely.

Virtual reality technology is becoming a powerful tool in surgical training as it promises a number of advantages: it allows repeated practice; it enables self-learning of surgical procedures without any time limit, and without risks to patients. Moreover the environment can simulate more than reality: for example, one can provide views inside the human body that are infeasible in practice. This can help students understand better the surgical procedures by allowing a mental visualization of internal structures and their spatial inter-relationship.

In order to build a useful VR-based training and assessment tool for the Percutaneous Rhizotomy Procedure (PRP), we need to understand the basis of the procedure and to identify the skills that the surgeons need. In addition, we need to identify which skills that can be best transferred into reality with current VR technologies. These issues will be addressed in detail in the following sections.

3.2 Medical Application

In this section, the medical background required in this study is described and task analysis is carried out. This will be followed by the discussion of technical requirements for the surgical training system.

3.2.1 The Treatment of Trigeminal Neuralgia

The trigeminal nerve in human heads consists of three large branches - see Figure 3.1. Trigeminal neuralgia can involve one or more of the three branches of the trigeminal nerve in the face and head. Tew et al. (1995) believe that it is caused by compression of the trigeminal root entry zone to the brainstem by vascular structures. It results in intractable facial pain involving the lower face and jaw, and sometimes around the nose and the area above the eye. People who suffer from this disease are normally older than 50 and it is more common in females. This is not a fatal disease but is one of the most painful afflictions to humanity. Years ago when no treatment existed, it caused some patients to succumb to despair and suicide because they were unable to endure the pain.

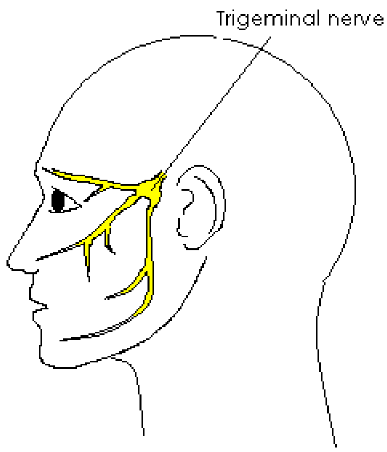


Figure 3.1: Trigeminal nerve. Illustration by Ruolong Chen.

Today, there are a number of ways to treat trigeminal neuralgia, including clinical and surgical treatment. The primary treatment is clinical and is often effective for a long period and relieves the pain within a very short time. However, some drugs used in the treatment may have side effects, such as bone marrow depression and liver damage, and/or inadequate pain control. Eventually, about 75% patients (Tew et al., 1995) need a further surgical intervention, as they will stop responding to the medicines or fail to receive long-term relief with medical therapy.

There are a variety of surgical procedures available for the treatment of trigeminal neuralgia. Each of these procedures has advantages and disadvantages. The choice of the surgical procedure for patients depends on the individual patient, e.g. the nature of the pain, or the health of the patient. At present, the most common techniques used include open skull surgery such as microvascular decompression, and a percutaneous treatment such as percutaneous stereotactic radiofrequency rhizotomy.

As a more invasive operation carries more risks to the patient, open skull treatment such as microvascular decompression surgery is often only an option for a young and healthy person. Because patients who suffer from trigeminal neuralgia are normally older than 50, there are a large number of patients who are recommended to undergo percutaneous treatment. More details of the percutaneous rhizotomy procedure will be discussed in the next section.

3.2.2 Percutaneous Rhizotomy Procedure

Percutaneous rhizotomy is a well-recognized treatment for trigeminal neuralgia (Mathews et al., 2000). A percutaneous rhizotomy procedure is performed with a needle directed with the aid of X-ray from the cheek to the nerve that triggers pain. Damage is inflicted on the nerve, and consequently the patient is freed from the pain with the sensation of numbness. Currently, a number of percutaneous procedures are available to release the pain. To damage the trigeminal nerve, each procedure requires a modality such as an electrode, glycerol or microballoon. All of these modalities have advantages in some aspects and disadvantages in others.

An example of such a procedure is percutaneous stereotactic radiofrequency (PSR) rhizotomy, which is the most selective and graded lesion (damage) among percutaneous techniques (Tew et al, 1995). The patient indicates the desired degree of numbness while the operation is being carried out. Figure 3.2 shows the PSR rhizotomy procedure, illustrated by Ruolong Chen, referring to the figure in (Mayfield Clinic, 2002). In general, the procedure is performed as follows. The patient is placed in the radiographic suite. The nurse administers anesthesia to a patient during PSR. The patient's arms are secured onto the table with a grounding wire. The surgeon is adjusting the current to destroy the pain-causing nerve fibers on a radiofrequency generator. The brain image from the fluoroscope is shown on the monitor.

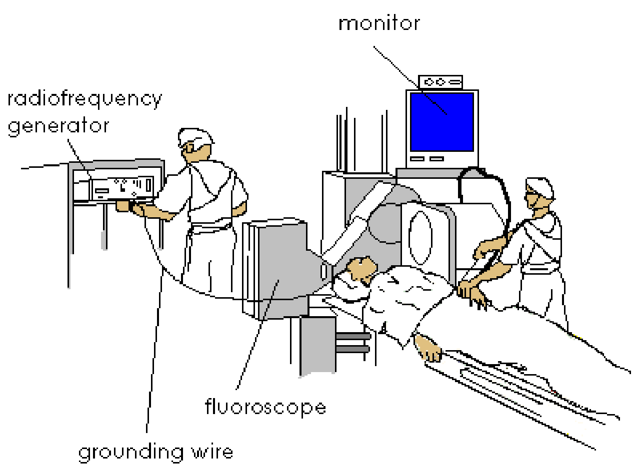


Figure 3.2: The PSR rhizotomy procedure

As part of the procedure is painful, so the patient is anaesthetised and falls asleep before the procedure. A spinal needle mounted with an electrode is then inserted from the cheek through a small opening in the base of the skull (foramen ovale) into the selected trigeminal nerve with a guiding device. Figure 3.3 shows the percutaneous stereotactic radiofrequency rhizotomy procedure: a needle placed in deep part of the trigeminal nerve. The figure was illustrated by Ruolong Chen, referring to the figure in (TN, 2002).

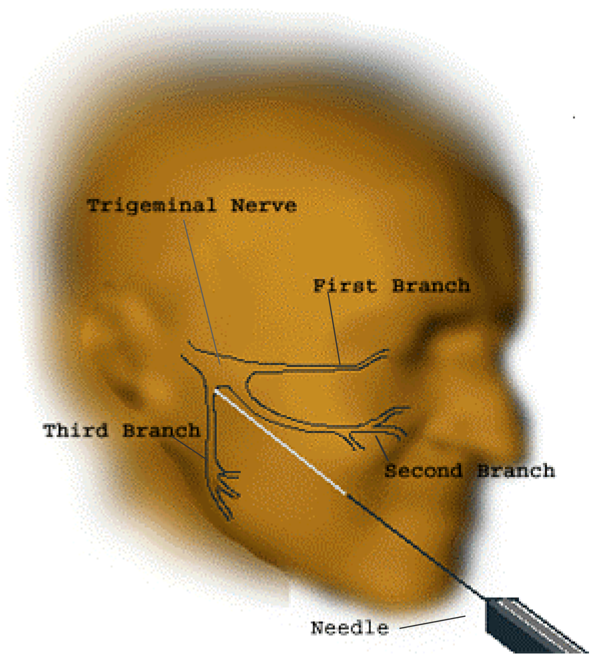


Figure 3.3: The percutaneous stereotactic radiofrequency rhizotomy procedure

During the placement of the needle, the patient is awakened to help the surgeon identify the location of the nerve that triggers the pain. This is achieved by passing a small electrical current through the needle to the face and causing slight tingling. The electrode in the needle is inserted so that the tingling occurs where the pain is triggered. Before the damage to the nerve is made, the patient is put back to sleep. A heating current through the electrode is applied to heat the location where damage is to be created. The patient has to be awakened again to help the surgeon check the amount of numbness (TN, 2002; Mayfield Clinic, 2002). This nerve damaging process is repeated until the desired amount of numbness is reached.

Other percutaneous rhizotomy procedures for the treatment of trigeminal neuralgia are similar in the way that the needle is positioned with the aid of X-rays to the trigeminal nerve through the foramen ovale from the cheek to lesion or destroy the nerve where the pain is produced. These techniques do not involve an intracranial operation. The procedures usually take about 1-2 hours. The pain is relieved immediately for most patients with these procedures. They can go home about two hours after the procedures.

Despite the number of advantages percutaneous rhizotomy treatment has over others, neurosurgeons are still restricted by the limitations of the current technology. The major difficulties they usually have to overcome are: the lack of visual cues, hand-eye coordination, location of the entrance point, and the trajectory of the needle movement. A further challenge is how to assess the performance of trainees. Hence, neurosurgeons need effective training methods and assessment tools to achieve a better level of performance before performing a real operation. In this research, we propose a solution to meet these needs – a web-based training simulator.

3.2.3 Requirements Analysis

A VR-based surgical training simulator is designed to train surgeons through participation. Clearly, the activities involve humans, computers and devices. A surgical procedure consists of a set of tasks. Performing a surgical procedure requires surgeons to use their skills, such as cognitive or psychomotor skills, to perform the surgical tasks. Each skill is acquired by using appropriate training modes. For example, text, 2D images and 3D anatomical models can all be used to help surgeons improve their spatial skills. Among those methods, the 3D anatomical model is the most efficient way, as the user does not have to make a mental effort to construct the 3D structure of the organ. It is necessary to perform a task analysis for a medical procedure and to define the skills needed for each task, in order to build functions in the system to perform the tasks and gain skills for them. Furthermore, each training mode must be defined for each task so that they can be well integrated to provide users an effective system.

Categorisation of Skills

The skills can be decomposed into basic components, which are essential to a skilled performance, e.g. automaticity, representational skill, problem-solving skill, decision-making skill, and strategic planning (Seamster et al., 1997). Each elementary skill is linked with one or more tasks. All of the basic skills are crisscrossed between the tasks.

Skills can be categorised according to the cognitive and practical activities (Seamster et al., 1997). The categories and their definitions are given in Table 3.1. A further analysis of the tasks is carried out in order to identify the skills to which the system will support.

Table 3.1: Definitions of skills

Skill Category	Definition
Psychomotor	Physical actions involved in the procedure.
Automated	Physical and cognitive activities performed rapidly and response to consistent stimuli or conditions quickly.
Perceptual	Visual or haptic sensory acquisition of information from the environment to support performance
Procedural	Constrained sequences of cognitive and physical activities which are performed in predictable situations.
Spatial	Spatial Cognition to guide the cognitive and physical activities.
Decision Making	Cognitive activities involved in choosing a better solution.

Medical Task and Skill Analysis

Surgical procedures consist of complex tasks. Analysing these complex tasks can be very difficult. The usual way to do it is to decompose the overall task into a series of simpler tasks (Seamster et al., 1997; Bro-Nielsen et al., 1999; El- Khalili, 1999). A further decomposition process may be needed if necessary. In this research, the selected procedure (percutaneous rhizotomy) is broken down into two parts: preparation and operation. Each part of the procedure is then divided into steps. Each step contains several tasks. Those tasks are analysed to determine the best training methods such as a 3D simulation, watching a video or a real operation, verbal instruction by an expert, text including photographs, or participation in a real operation, so that the skills can be transferred to reality efficiently. In a 3D simulation, there is an overhead associated with learning how to use the simulation. If the simulation contains too many, and too detailed tasks, then there is a danger that the use of the simulation becomes overly complex. The trainee will be distracted from learning the tasks because of the need to learn how to use the software. Thus only key tasks are identified in the work and incorporated in the simulation. In addition, it is relatively easy to learn most steps of a procedure by watching and participating (Tendick et al., 2000). One needs to decide which training mode is best applied to the skills for each task.

❖ Preparation

This part is divided into three steps:

Step 1: Knowledge acquisition.

It is not possible to develop skills without knowledge. The knowledge of the procedure and the knowledge of the pertinent anatomy should be gained before performing a real operation. The knowledge of the procedure is a component of procedural skill. An easy way to gain the knowledge is by reading or listening to an instruction or watching a video or real operation. Similarly, the knowledge of the pertinent anatomy is a component of spatial skill. This can be mentally constructed by 2D graphics or a simulated environment with a 3D anatomical model.

Step 2: Making anatomical landmarks.

In order to locate the needle position, three anatomical landmarks are chosen on the face of the patient. They are marked with a pen and then washed with antiseptic. This task requires decision-making, and psychomotor skills. These skills are better transferred to reality by practising in a virtual environment, rather than the operating theatre. A 3D simulation provides a practical environment for users to practise their decision-making, and psychomotor skill for making appropriate anatomical landmarks without the involvement of experts.

Step 3: Preparing patient.

Atropine is administered in order to prevent bradycardia and to reduce oral secretions. Other tasks in preparing a patient include: checking patient’s blood pressure and heart rate with the assistance of a nurse, and so on. An automated skill and a procedural skill are required for these tasks. Both skills for the tasks can be learnt through watching a video or real procedure, and verbal instruction first to build the basic knowledge, and then enhanced by participation.

The skills required for the tasks and the better choice of the training modes are summarised in Fig. 3.4.

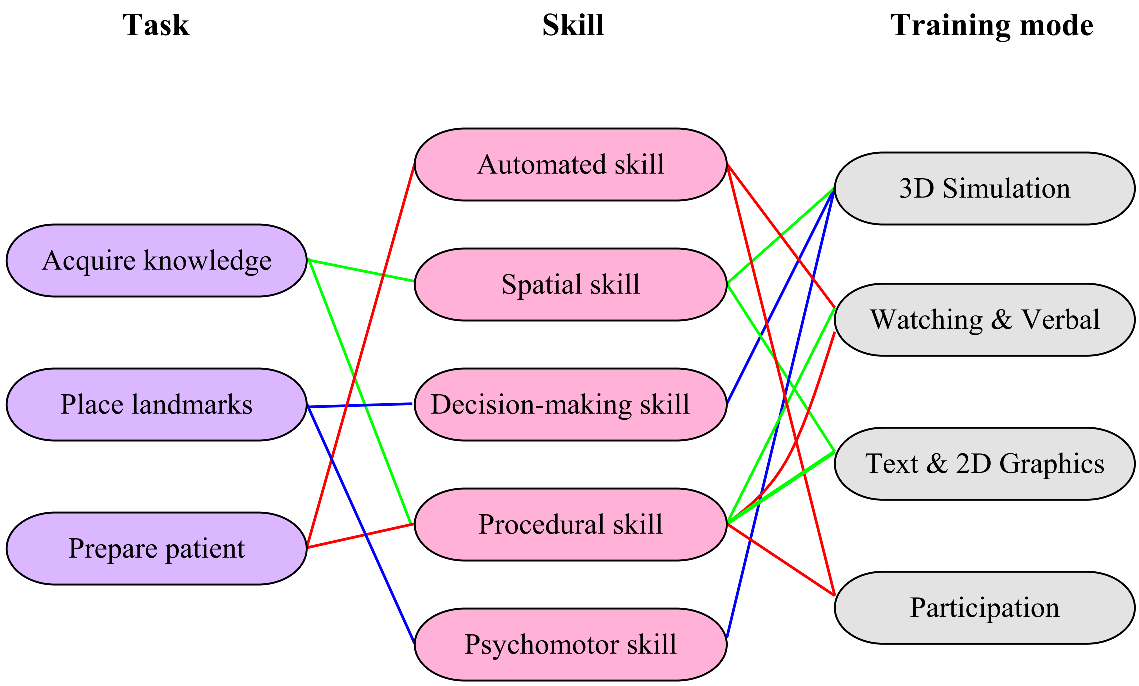


Figure 3.4: Task analysis in preparation

❖ Operation

In Section 3.2.2, the detailed process has been discussed. This part of the procedure is divided into three steps:

Step 1: Insertion of the needle

Once the anatomical landmarks are set, the entry point of the needle is then defined. The insertion of the needle requires psychomotor skill to perform the manipulation of the needle. Psychomotor skill is usually gained by practising. However, it is dangerous if the trainee practises on patients for this task. Therefore, a 3D simulation provides an alternative way to train psychomotor skills in this step.

The position of the needle is corrected by using guidance from X-ray imaging during the operation. Surgeons need to mentally construct a 3D picture with depth information from the 2D X-ray image so that a correct movement can be carried out. As far as the harmful effect of X-rays to the patient is concerned, the more often X-rays are taken during the surgery, the more dangerous it is to the patient and the longer the operation takes. Hence, spatial skills are required for this task. A 3D simulation is able to represent the location of the needle and the surrounding structure. In addition, the relation between the needle and the skull can be discovered during the insertion movement with a 3D simulator, and it therefore helps a trainee to gain spatial skills for this task.

As the electrode in the needle is advanced, the entrance of the needle into the opening of the lower skull is signalled by a wince and a brief contraction of the masseter muscle. Neurosurgeons can sense it based on their haptic sensation, a perceptual skill. This feedback indicates a correct position through the opening. Before any further movement of the electrode, a lateral image is taken to confirm a proper placement of the electrode. Again, spatial skills are essential for this task. By practising on a 3D environment with haptic media, this can help trainees to acquire perceptual skill and spatial skill.

Step 2: Identifying the source of the pain

Once the needle is entered into the opening, a final placement of the electrode tip is carried out based on the patient's response to electrical simulation. This step involves a regressive process and needs the cooperation of the patient to find where the source of the pain is. Decision-making, psychomotor, automated, and procedural skills are all useful to decide the location according to the response of the patient. These are best trained with participation in real surgery with the patient's reaction.

Step 3: Damaging the pain-causing nerve fibres

This step again needs the cooperation of the patient. The damage is repeated and continued until the desired effect is achieved. Decision-making, psychomotor, procedural, and automated skills are necessary to decide how much the damage is, according to the patient's response, and how to produce the damage. Participation in real surgery is important for the trainee to build these skills, as patient involvement is essential. The skills required for the tasks and the choice of the training modes are summarised in Fig. 3.5.

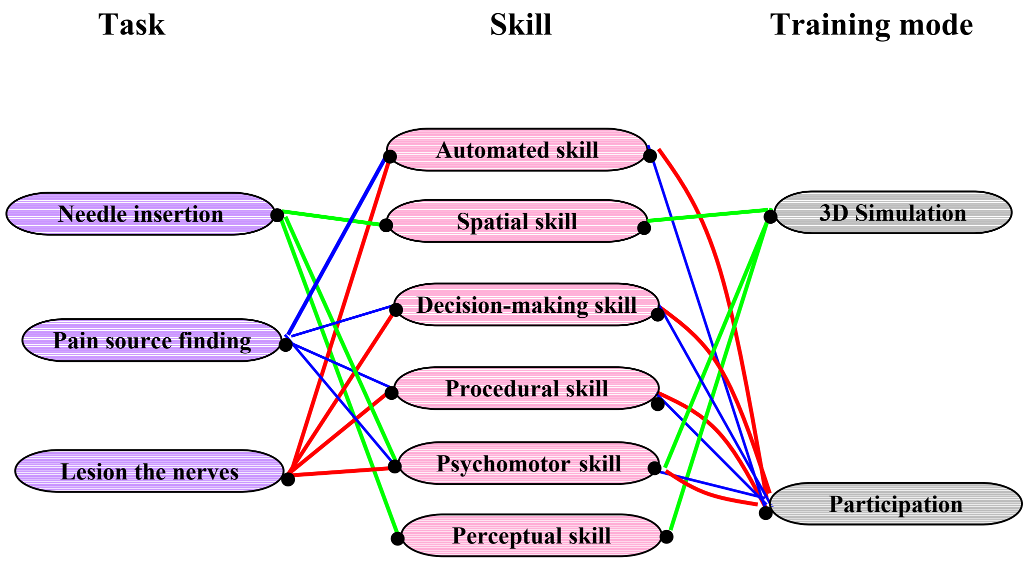


Figure 3.5: Task analysis in operation

In Fig. 3.4 and Fig. 3.5, the skills for each task that need to be trained in a VR environment are pointed out. Some traditional training methods, such as text and graphics, can be incorporated into a VR environment. These skills are filtered from the two figures (3.4 and 3.5), in terms of the training modes in which we are interested. Together, these are summarised in Table 3.2:

Table 3.2: Tasks to be achieved through VR tool

Tasks	Skills	3D Simulation	Text & Graphics
Acquiring Knowledge	Spatial Skill		*
	Procedural Skill		*
Placing Landmarks	Decision Making Skill	*	
	Psychomotor Skill	*	
Inserting Needle	Spatial Skill	*	
	Psychomotor Skill	*	
	Perceptual Skill	*	

Neurosurgeons need alternative training tools to meet the new challenges in surgery, and quantitative assessment tools to measure their performance. In a collaboration between computer scientists and neurosurgeons, this research is directed toward the study of how to use VR technologies to provide a training platform and assessment tool for neurosurgeons in one of the popular procedures, percutaneous rhizotomy. In particular, this work focuses on certain tasks of the procedure listed in Table 3.2. In the next section, a system design process is carried out based on the task analysis for the procedure as described earlier.

3.3 Percutaneous Rhizotomy Procedure System

The criteria for surgical simulators have been identified in (Dawson et al., 1998): realistic, affordable and validated. In fact, it is sometimes difficult to satisfy all of these criteria with the current technology. For instance, a realistic simulator requires realistic anatomical models, realistic behaviours and high fidelity interfaces. With today's technology, these must be achieved by sacrificing the low cost criterion - as a powerful machine is needed to display anatomical models with high resolution and to represent their realistic dynamic behaviour. This, together with the high fidelity interface, is very expensive today. In addition to the criteria listed, according to the analysis as described earlier, accessibility is also an important criterion – the training tools should ideally be available anywhere, any time - and it is the first we consider in this study. In the design phase, all of these criteria are considered and compromised in order to achieve the goal set in this study.

3.3.1 System Architecture Design

Using the web as a training environment will fulfil two criteria: accessibility and affordability. However, it will raise a new challenge: how do we provide sufficient realism in the system. Three main factors affect the realism in a surgical training simulation: the anatomical models that are used in the simulation, the behaviour in response to the users action, and real time performance during the interaction. The architectural design for a web-based application can directly affect the performance.

As mentioned earlier, the approaches for Web applications fall into two groups: client-based and server-based. The decision on which of the approaches should be adopted for the application is initially dependent on the requirement of the actual application and the goal which we are aiming to achieve. There are several reasons why the client-based design should be adopted:

- Less dependence on the server: Once the application and data are downloaded to the client side, the client can execute the entire application without any further involvement of the server. Thus if the server goes down, the process can continue.
- Less dependence on the computer network: Because there is no involvement of a server during operation, the performance will not be affected by any network delay. In addition, with a server-based solution there are often issues in synchronising the processing of the client front-end and the server back-end.

Under this design, there are several things that need to be considered:

- Considering the demand on the client, the size of the data and the computational costs should not be very large.
- In a client-based solution, the developer has less control over the software that is used (for example, which browser, which graphics viewer, and so on). It becomes difficult to cope with the complexity of multiple versions of software components. A pragmatic solution is to ensure the system functions correctly with current versions of a specified combination of products.
- In a client-based solution, it is harder to support collaborative activities. For assessment purposes, some persistent records of a trainee's

performance are required and so we shall need to make some limited use of a server for this purpose.

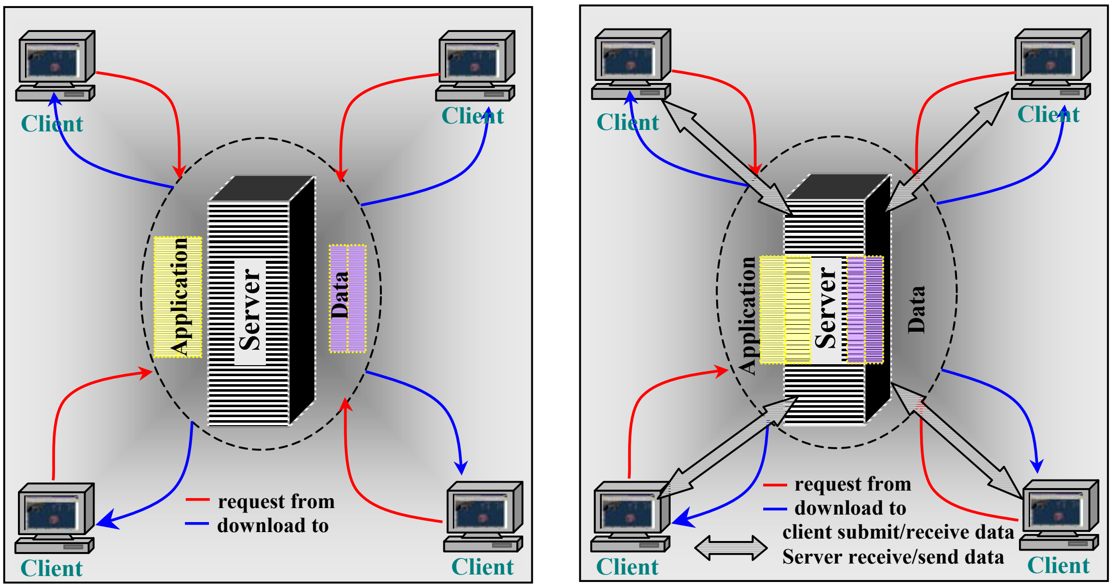


Figure 3.6: Left: Client-based architecture, Right: Server-based architecture

In contrast, the server-based approach should be adopted when it is necessary to distribute the computation. This is needed when we are working with large data sets, when the computational costs are high, or when we need to support synchronous collaboration. With this approach, the application is split into two parts. One part is allocated to the client site and the other is allocated to the server side. How the application components are distributed will affect the system performance, and even though care is taken to optimally configure the software allocation across resources, performance can still be affected by network delay. Hence, the client-based architecture is preferable for applications where the interactive rate is an important factor.

In this study, real time interaction is important for the realistic performance. Therefore, a client-based approach is adopted. In Table 3.2, the skills for the tasks in the procedure that can be supported by practical training through VR tools, and the skills that need a media, such as text and graphics, have been identified. In each design phase, it is necessary to consider an appropriate way for each skill so that it can be transferred to reality efficiently.

3.3.2 System Overview

To provide neurosurgeons with a web-based surgical training tool, the enabling technology on the Web must be deployed. At present, the most commonly used technologies for web-based virtual reality are VRML and Java 3D.

VRML is a 3D file format that can be embedded on the Web. It was first released in 1995 but only static worlds could be displayed with the first version. A much improved version was released in 1997, where interactive behaviour, such as touching and dragging, together with other advanced features were added. Just like Java, VRML is platform independent. It is an attractive tool when these two work together – extending the VRML and building platform independent applications. To put them together, two options can be chosen: inserting Java in VRML Script nodes, or accessing a VRML browser from an applet using the Java EAI (External Authoring Interface). Java EAI allows Java Applets to access the VRML worlds so that the functions of VRML can be extended to model complex interactions of objects, and support the collaboration between VRML worlds.

There are a few different VRML browsers which are available at present. The most commonly used are: Cosmo Player (CosmoPlayer, 2002), Blaxxun Contact (Blaxxun Contact, 2002) and Cortona (Cortona, 2002). These now all support Java EAI.

An alternative enabling technology on the Web is Java 3D. Java 3D is part of the JavaMedia suite of APIs, which supports the building of three-dimensional graphics applications and applets. Because applications and applets written using Java 3D API have access to the entire set of Java classes, they will run on a wide range of platforms and integrate with the Web (Java 3D, 2002). The Java3D scene graph model shares some common features with VRML. It provides support for runtime loaders so that a wide variety of file formats, such as VRML, can be imported to Java3D. Moreover, it supports stereo viewing and VR input devices. However, at the time of the study, a number of bugs in Java3D were identified, which make the implementation of the tasks needed in the applications impossible. For example, the returned value from the method *PickIntersection.getPointCoordinates* is wrong, so the correct position of the selected point cannot be obtained from the picking utility from the early version of Java 3D. Although some of the bugs had been corrected in a later version of Java3D, it is still not mature enough for some applications.

In this research, VRML combined with Java EAI (External Authoring Interface) is therefore used to render the 3D objects and to handle their interaction. There are three basic components that should be included in a surgical simulation: 3D geometric models (for example, instruments and anatomy), interactive modelling (for example, response to collisions) and user interface design (for example, choice of interaction styles and physical input devices). In the following sections, we will describe how we designed each of these three components in order to develop a Web-based surgical simulation system to help the neurosurgeons to obtain the skills require during the PRP procedure.

3.3.3 Geometric Modelling

As a VR-based surgical training tool, the visual display is an essential component. This is represented by a set of objects, mostly 3D objects, in the virtual environment. A 3D object can be defined as either surface representation or volume representation. Compared to the surface representation, a volume representation requires more size to store interior data of the object. As mentioned earlier, the size of the data is crucial in the client-based environment, and moreover surface representation is assumed in VRML. This leads to the adopting of the surface representation for the objects displayed in the virtual environment.

Spatial skill is important in the PRP procedure. A realistic 3D anatomical model helps a trainee to acquire this skill. In order to model the head and skull realistically with a surface representation paradigm, real human data has been utilized. This data was generated by Manchester Visualization Centre by using a laser scanner – the Rapid 3D Digitizer Model 15, from CyberWare. The software supports VRML output and produces extremely large 3D data sets. A highly accurate polygon reduction algorithm was used to reduce the size of the data (John et al., 1999). In this research, a further polygon reduction was carried out using the modelling tool (Cosmo Worlds) in order to speed up the loading time from the Internet.

As described earlier, the target point for a needle is an opening in the skull called the foramen ovale. There are two of these, one on each side of the head, at the base of the skull. These are, however, not included in the model of the skull. These two holes were added to the skull model manually. The size was scaled and placed into the appropriate position in the skull according to the suggestion of the neurosurgeons (6/7mm wide, the height is 3mm).

The needle and other objects, such as the interface widgets described below, are not important for the spatial skill. These models are therefore kept as simple as possible to reduce the size of the system. Again, the modelling tool, Cosmo Worlds, was used to model some of the objects, scale the models and place them at their correct position and orientation.

3.3.4 Interface Design

Interaction – choice of input device

Before an interface design for the virtual environment can be carried out, the type of interaction must be defined. The use of different devices to perform the interaction will affect the degree of realism. As mentioned earlier, the degree of realism must be balanced with other factors, such as costs and ease of use of the tool. The requirement for the system to operate in a Web environment, with standard browser tools, further restricts the style of interaction.

To perform the tasks of making landmarks and inserting the needle, listed in the task analysis table (Table 3.2), we need to define the location of each landmark on the model and to manipulate the needle. In addition, manipulation of the viewpoint is needed to acquire spatial skill, and gain a good understanding of the 3D relationship between the needle and the skull. This can be achieved by using 2D/3D input/output devices. There is a wide range of 2D/3D input/output devices now available, but no device is supported by standard browsers directly. Recently, a force-feedback mouse, which is a two-degree of freedom device from Immersion Corporation (Immersion Corporation, 2002), has been used in the system described in (Riding et al., 2001). It provides haptic effects, including the feeling of forces (opposing or supporting force) in arbitrary directions, feelings of texture, and vibrations, and allows a user interacting indirectly with the VRML scene. At present, there is no evidence whether this low performance VR can provide an effective platform for the user. Therefore, in this research, the 2D standard mouse is utilized to allow the user to interact with virtual objects though the user interface.

User Interface Design

The primary goal of the user interface in this design is to allow the user to perform surgical tasks listed in Table 3.2 easily and efficiently. In order to achieve this goal, the design focuses on the specific tasks. The appropriate methods are then chosen to perform those tasks through the user interface. In Figure 3.7, three functions are set. These functions are broken down into sub-functions. The components of the interface are then defined for them to perform the surgical tasks.

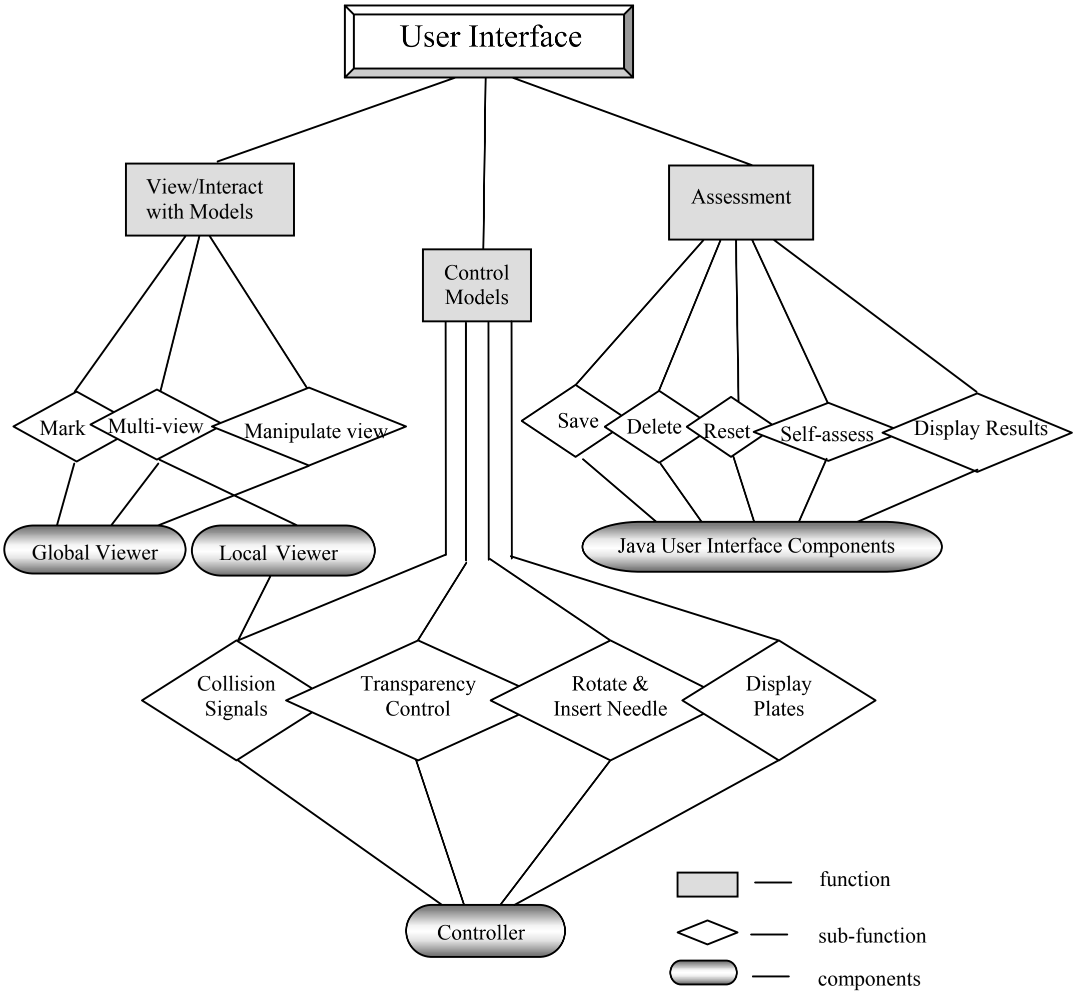


Figure 3.7: The User Interface design

The *interface* shown in Figure 3.8 is provided by a web browser, where three VRML worlds and one applet are embedded into a HTML page. These three VRML worlds are a local viewer, a global viewer and a controller. In order to make it easy to use, most interactions are gathered in the controller.

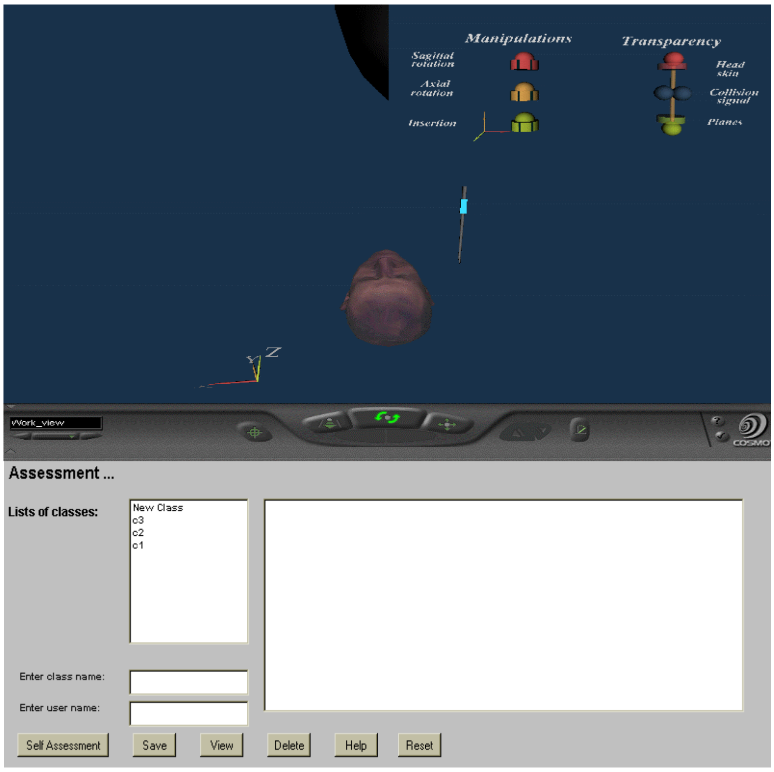


Figure 3.8: The User Interface

The *local viewer* (see Figure 3.10 on the left) shows the view from the position of the tip of the needle in the global viewer. When the needle in the global viewer moves, the local viewer will change its state according to the movement of the needle. This view of course is not available to the surgeon in reality. It is simply to aid the trainee's conceptualisation of the procedure, and help him during training to achieve the correct result and gain the decision-making skill. This viewer also plays an important role in the collision detection between the needle and the head in the global viewer, as will be explained in the next section. This view is "frozen" so that the user cannot manipulate the view from the standard VRML browser interface controls (another example of minimizing complexity of the interface).

The *global viewer* (see Figure 3.9) shows the view from an external viewpoint (the surgeon's eye view). This view represents the objects involved in the surgery: the instruments and the patient's head, and their interactions. The head at "On Table" position is initially displayed when the browser is loaded. The spatial skill can be enhanced by allowing the user to manipulate the view from an arbitrary angle and select different views from the standard VRML browser interface controls. The psychomotor skill is gained by clicking on the face interactively to obtain the landmarks. When the first landmarks are placed, the needle is located at the position of the landmark automatically. (This is an example where we avoid complexity in the operation of the simulator – by moving the needle to the correct starting position automatically – so that the trainee can focus on the key subsequent tasks of orienting the needle correctly and inserting it to the correct depth.)

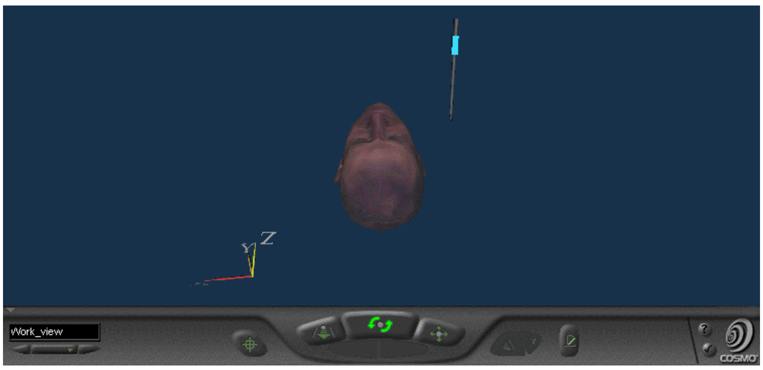


Figure 3.9: The Global Viewer

The *controller* (see Figure 3.10 on the right) displays the widgets, which are used to manipulate the movement of the needle, control the visibility of some features in the global viewer, and give some feedback to the user when the needle collides with the skull or is inserted into the foramen ovale. The user's mouse inputs are mapped into the widgets individually to determine the amount of rotation and insertion. Corresponding transformations are then applied: to the needle in the global view, and the viewpoint in the local view. In order to help the trainee manipulate the needle correctly, a number of special effects have been included. Firstly, the skin may be made transparent so that the skull itself is visible. Secondly, the three reference planes, namely the imaginary axial, coronal and saggital planes, through the ideal landmark points respectively, can be switched from invisible to visible. The needle starts at the first landmark point on the cheek in the imaginary axial plane, 2.5cm from the angle of the mouth. It may need to be adjusted in the imaginary axial plane so that it points at the intersection of the three reference planes. The needle can then be rotated vertically and directed into the foramen ovale by aiming the needle toward the intersection of the coronal and saggital plane. Trainees are taught to visualise these planes when learning the operation. These visibility features are used to help the user locate the needle correctly and gain the decision-making skill. Again, this view is "frozen" so that the user cannot manipulate this view from the standard VRML browser interface controls. In order to be able to mimic the steps involved in a real procedure, the movement of the needle is constrained by disabling the rotation capability after the insertion movement is chosen. The communication between these VRML worlds is achieved by the Java External Authoring Interface (EAI).

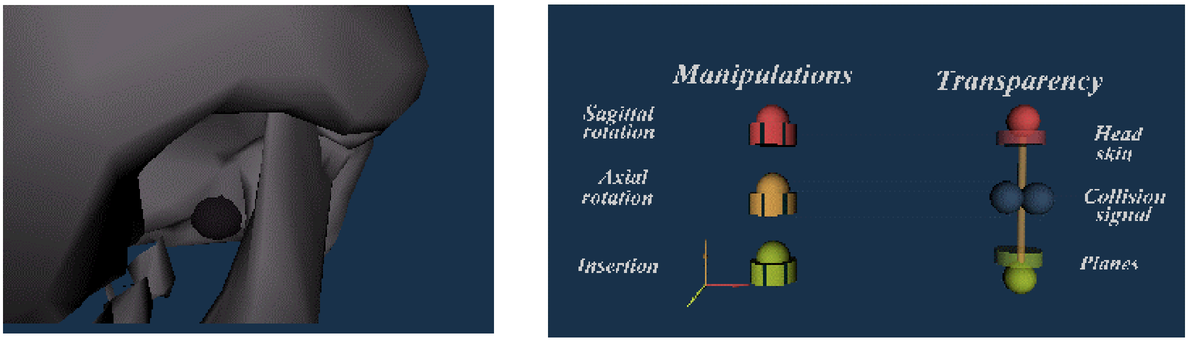


Figure 3.10: Left: the Local Viewer; Right: the Controller

Assessment

In the UK, it is the duty of the trainer to identify and remedy defects in the knowledge and performance of a trainee during surgical training to ensure progress at each level of training (Joint Committee on Higher Surgical Training, 1999). In order to achieve this task, there is a need to monitor the progress of trainees throughout the training programme.

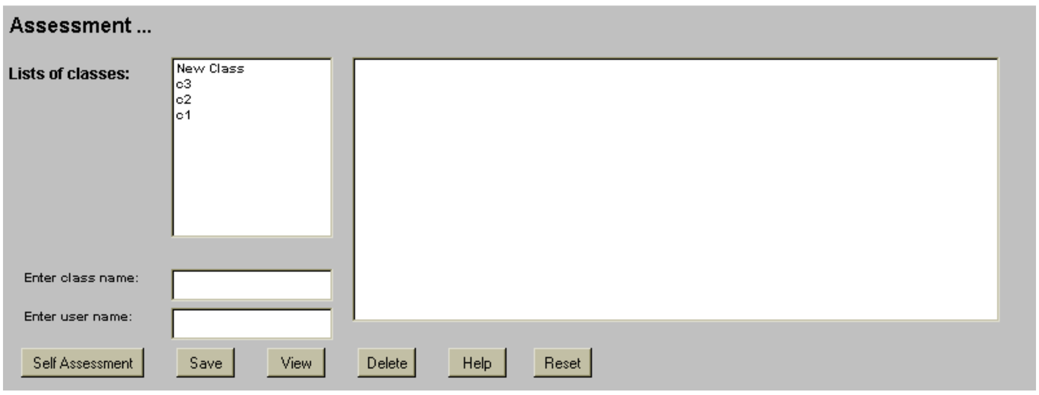


Figure 3.11: The Assessment tool

Traditional evaluation methods of the progress of a trainee are mainly based on observation, which is potentially unreliable. Therefore, a number of objective methods have been developed to assess a trainee's performance, and these have been used in virtual environments. Examples include using an automated high-frequency tool to provide tracking and a postural measurement system to identify kinematic features of specific surgical actions of laparoscopic cholecystectomies (McBeth et al., 2002); recording the time taken to complete a task and the trajectory of the tip of the tool for assessing laparoscopic surgical skills (Payandeh et al., 2002); measuring the force, time taken, and the length and straightness of the stitch during the performing of suturing tasks. These applications provide appropriate quantitative performance feedback to the trainers and trainees so that the trainers can identify any lack of progress and give appropriate advice quickly to the trainees.

In this application, an assessment tool, shown in Figure 3.11, is integrated into the surgical training system. The key metrics of performance in this training system have been defined. Two metrics are used to assess the performance:

- ❖ Measuring the position of each landmark

  When performing the marking task, three landmarks are placed on the patient's face in order to help locate the position of the needle: (1) a point 2.5 cm lateral to the oral commissure (corner of the mouth), (2) a point beneath the medial aspect of the pupil, and (3) a point 3 cm anterior to the external auditory meatus. Point (1) is the point at which the needle penetrates the skin of the jaw, and the other two points indicate the site of the foramen ovale (Tew et al., 1995). As explained earlier, these two points are associated with coronal and saggital planes, whose intersection is used to guide the needle towards the foramen ovale.

According to these criteria described above, two measurements for each point are important for determining whether the marked point is correct: the length between the marked point to a specified point on the face and the distance from the marked point to a specified plane. For point (1), an axial plane passing the corner of the mouth is defined. The length from the point (1) to the corner of the mouth must be 2.5 cm, and this point has to lie on the axial plane. The longer the distance from the point to the axial plane is, the worse the result would be. For the point (2), a saggital plane is defined. The length from point (2) to the medial aspect of the pupil must be zero, and this point has to lie on the saggital plane. The longer the distance from the point to the saggital plane, the worse the result. For point (3), an axial plane is defined. The length from the point (3) to the tragus must be 3 cm, and this point has to lie on the axial plane. The longer the distance from the point to the axial plane, the worse the result. The metrics are shown in Figure 3.12 and Figure 3.13.

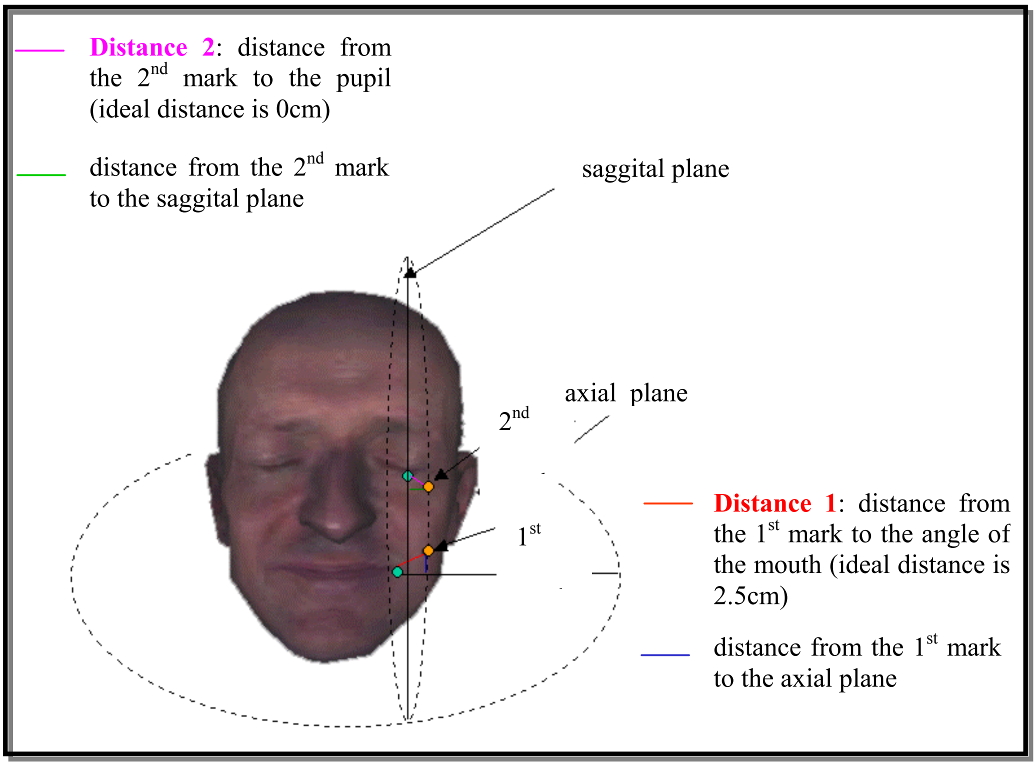


Figure 3.12: The measurement of the point(1) and (2).

❖ Measuring the target point of the needle

Once the landmarks are defined, the needle is adjusted and then directed into the medial portion of the foramen ovale. When the tip of the needle hits the skull or the target – the foramen ovale - the distance between the tip of the needle and the centre of the foramen ovale, on the same side as the entry point, is measured. This measurement is important as it can give an indication of how far or close the adjustment and insertion of the needle is to being successful. The metrics are shown on Figure 3.14.

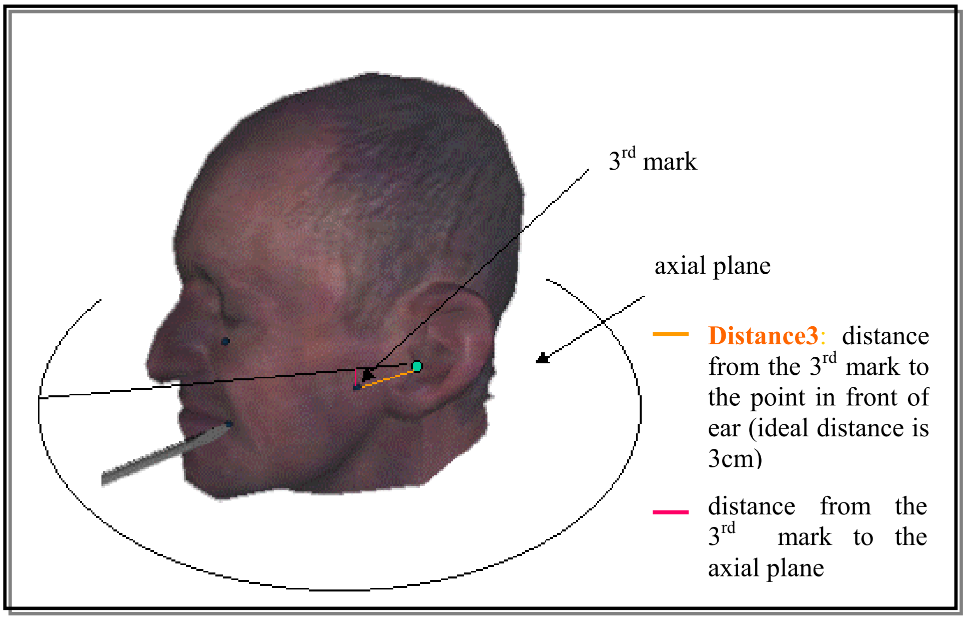


Figure 3.13: The measurement of the point (3).

As mentioned earlier, the trainer needs to assess the trainee at each level of the training program so that they can identify the defects and give the appropriate advice to the trainee during the training. It would be useful if the training system could record and retrieve the metrics defined above when each task is performed. Not only can the trainers benefit from it, as these metrics are important information for monitoring the performance of the trainee, but also the trainees, as these measurements can be analysed by themselves and therefore they can make a better decision in the next attempt.

In order to facilitate this function, an assessment tool is embedded into the simulator. Once the three landmarks are placed on the face in the simulator, the user can click on the *Self Assessment* button and the measurements are displayed in the text field (see Figure 3.11). After completing the procedure, the user can select the *Save* button to save the results into an appropriate group with a specific user name. The results can be viewed later on or deleted from the group. If the *Help* button is chosen, the documentation, explaining how to use this assessment element and how it works, is displayed.

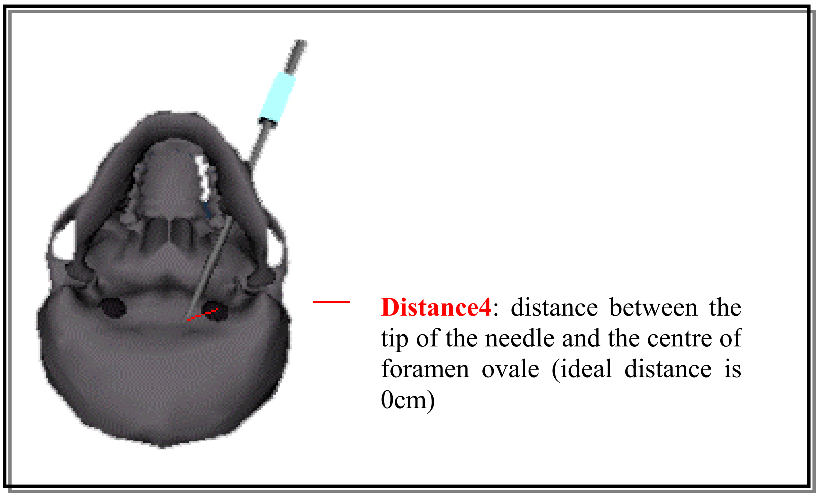


Figure 3.14: The measurement of the tip of the needle to the target point

3.3.5 Object Manipulation

The movement of the needle involves rotations and insertion in 3D space. In order to support the psychomotor skill in performing needle manipulation, the simulator provides a virtual needle so that the user can manipulate it in the virtual environment. As indicated earlier, the interactions are mainly congregated in the *controller* for ease of use. Thus, the insertion and rotation movements are mapped from the *controller* widgets to the needle in the *global viewer* as well as the *local viewer* through a Java Applet. The "SphereSensor" node facilitates the rotation of the objects about an arbitrary axis. In this application, however, we require first a horizontal rotation within the axial plane, and then a vertical rotation if it is necessary. The “SphereSensor”, in practice, is not well suited to handling rotation precisely about a specific axis. Therefore, we prefer the idea of rotating the needle about the different axes by using separate cylinders, each of which controls the rotation about one axis. They are manipulated by “PlaneSensor” nodes in VRML using a mouse, and the displacement generated from the “PlaneSensor” is then transferred into the orientation of the needle movement in the *global viewer*. For reasons of consistency, the insertion of the needle is designed in a similar way to the rotation – manipulating the “PlaneSensor” attached on a cylinder using a mouse, and the displacement is transferred into the insertion of the needle in the *global viewer* and the viewpoint of the *local viewer*. To concatenate rotations, quaternions are used.

Quaternions are often used in computer graphics to represent rotations. There are a three-element vector and a number corresponding to the angle, in a similar way to VRML for representing rotation. It is easier to accumulate than the form in VRML. The expression for a quaternion is:

$$\mathrm{q} = w + xi + yj + zk = [w, x, y, z] = (s, \mathbf{v})$$

where w, x, y and z are real numbers; $s = w$, $\mathbf{v} = [x, y, z]$.

It is an extension of complex numbers. Instead of just i, it has three imaginary terms, i, j, k. The calculation, such as addition, subtraction or multiplication, is in much the same way as complex numbers. When it is used to represent rotations, the form of computing a rotation about the unit vector, **u,** by an angle theta is as:

$$q = (s, \mathbf{v}) \qquad (3.1)$$

where, $s = \cos(\theta/2)$ and $\mathbf{v} = \mathbf{u} * \sin(\theta/2)$, $\mathbf{u}$ is the vector of a rotation.

Before calculating the rotation, the rotation form in VRML needs to be converted into a quaternion form using the formula (3.1). When concatenated rotations are required, a multiplication of these rotations in the quaternion form (Quaternion, 2002) is performed.

Computation of the product of two quaternions is given as follows:

$$\begin{aligned}
&q_1 = w_1 + x_1 i + y_1 j + z_1 k \\
&q_2 = w_2 + x_2 i + y_2 j + z_2 k \\
&q_1 * q_2 = (w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\
&(w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2) i + \\
&(w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2) j + \\
&(w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2) k
\end{aligned}$$

An alternative way to do the accumulation of the rotations is to use the VRML97 facilities within a Script node to capture the input data and to accumulate the total rotation, and then send the result out.

3.3.6 Collision Detection

Many applications involve multiple objects that interact with each other. In a realistic simulation, collision detection is essential for those applications. This is not only because it provides realistic output of the interaction for analysis, spatial reasoning and visualization, but also because it gives useful visual clues to represent a perceptual sensation. As mentioned earlier, in the PRP procedure, when the needle is inserted into the foramen ovale, a wince sensation can be felt. This means that the needle has been successfully inserted, and the surgeon can move onto the next step of the procedure. As we have noticed, the VRML 2.0 specification only supports the collision detection between a user avatar's viewpoint and the scene geometry. The collision between objects in the scene is not supported (simply for performance reasons). Therefore, there is a need to develop an alternative solution to fulfil this requirement.

Many collision detection algorithms have been proposed in recent years (UNC Gamma Research Group, 2002; Hudson et al., 1997). Because of the computationally intensive nature of the algorithms, many of them focus on accelerated techniques, mainly using hierarchical representation (Palmer, 2002; Gottschalk et al., 1996) and spatial decomposition (Baker, 2002). With the hierarchical representation, the objects in the scene are approximated by bounding volumes. The inner-object is divided into subsets and then approximated by a number of sub-bounding volumes. This continues until the subset contains one or a small number of polygons. The collision detection is carried out from the root level. If it is intersected, the second level is checked. This continues until the intersected polygon is identified. The spatial decomposition works in a similar manner except that instead of approximating the objects with bounding volumes, it partitions the space occupied by the objects into subspaces using decomposition techniques. In this way, the number of pairs of objects and pairs of primitives that need to be examined for contact or intersection are drastically reduced.

The choice of collision detection algorithm depends on the model representation, the desired query type (for example, which points are colliding, or which triangular facets are colliding), and the simulation environment (Lin et al., 1998). The information, such as the contact and intersection position, and time estimation of the next collision, are required for different applications. For example, in some applications, such as the system developed by John and Riding (1999), it is only important to decide whether or not the object is collided with another. In other applications, such as the system developed by El-Khalili and Brodlie (1999), deciding which pair of primitives collided is essential for collision response calculation.

We need to extend VRML in order to provide object-object collision detection. The V-COLLIDE library has demonstrated that accelerated collision detection for VRML can be achieved. One approach is to develop a special VRML browser by interfacing to the V-COLLIDE library (Hudson, 1997). The disadvantage of this approach is the special browser must be maintained and distributed to all users. El-Khalili used a server-based architecture for the simulator, thus the collision detection library (V-COLLIDE) can be placed on the server side. The approach is not suitable for a client-based application. John and Riding (1999) developed a client-side approach, which uses standard browsers and provides a simple collision detection facility, to handle object-object collision. An off-line pre-processing is needed for runtime efficiency.

In our application, the situation is different from (John and Riding, 1999): the objects involved in the collision are more complex, there are several objects (skull, jaw, two foramen ovale, and needle) and we need to distinguish which object (skull or foramen ovale) the needle is colliding with. A needle-eye view approach is exploited in the simulator, and we use that for collision detection, in the following way.

In the *local viewer*, the position of the viewpoint corresponds to the position at the tip of the needle in the *global viewer*. Since VRML supports collision detection between the viewpoint and the objects in the scene, when the collision happens in the *local viewer* (between the viewpoint and the objects), the message is passed to the *global viewer* and the *controller* through the Java EAI. The *controller* browser signals to the user with coloured lights to give feedback of success (green light) or failure (red light). The user can decide which action is to be chosen in the meantime. For example, if the needle collides with the skull base the user can check how far it is from the foramen ovale from both *local viewer* and *global viewer*. The entry point can then be adjusted in the next practice. It therefore supports the decision-making skill.

The position captured in the tip of the needle cannot be directly passed to the position of the local viewpoint as they are in different co-ordinate systems. Thus the transformation is calculated to bring the positions into the same co-ordinate system.

The simulation is able to identify collision with the different objects. When it receives the collision message from the local viewer, it gives appropriate signals to the different collisions. In the simulation, for example, the entrance of the needle into the skull base is signalled by a red light. When the user continues inserting the needle, the simulation reverses the motion of the needle so that it appears to bounce back from the skull. On the other hand, the entry of the needle into the foramen ovale is signalled by a green light and a compliment!

A major advantage of web-based VR is that the techniques generalise to a whole range of surgical simulations. Thus we have been able to use exactly the same approach as described above for neurosurgery, to develop a shoulder arthroscopy simulator as shown in Figure 3.15 - where again collision detection, and the view from the scope, are fundamental.

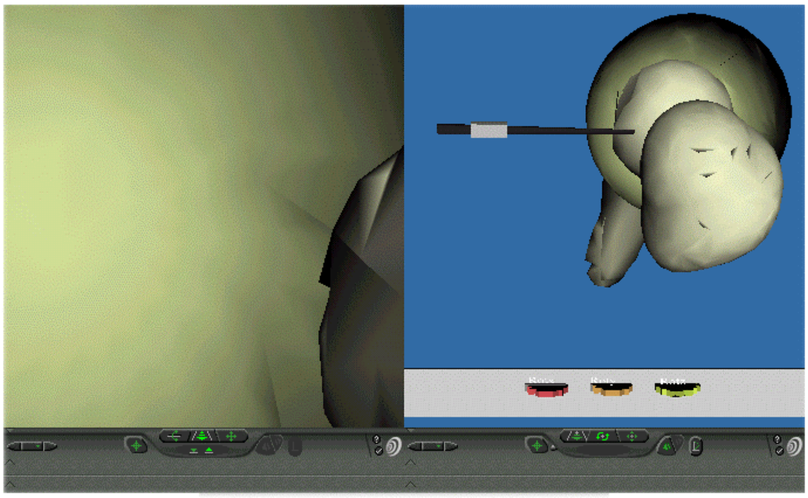


Figure 3.15: Arthroscopy simulator – implemented by Ying Li. The model of the bone was provided by Erik Beaumont

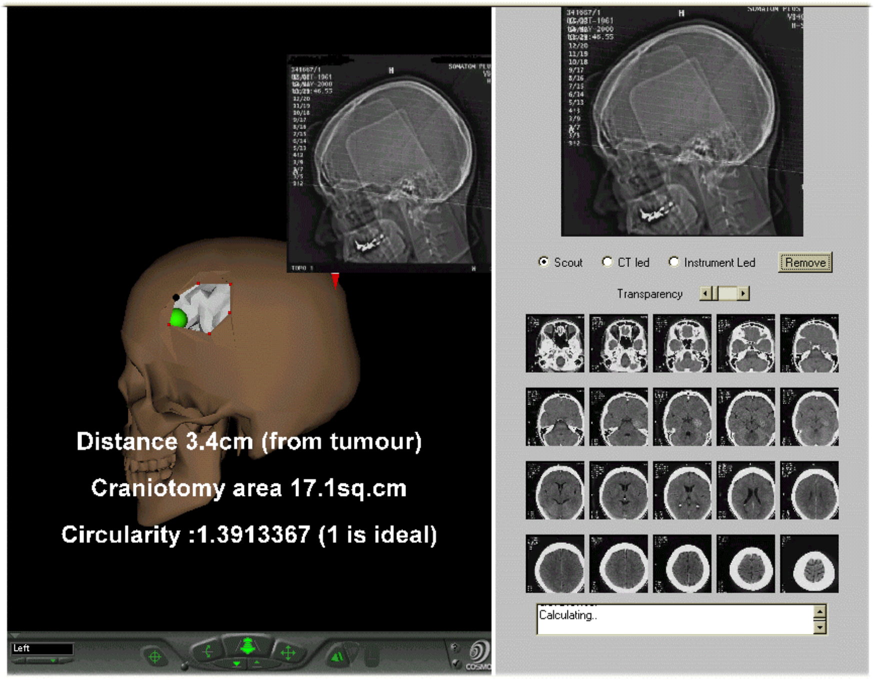


Figure 3.16: Craniotomy simulator – implemented by Patrick Sim, Ying Li, and Sam Underhill.

3.4 Documentation

Documentation is useful for background learning. The background knowledge of the procedure and the knowledge of the spatial relation, as shown in the Table 3.2, are required before any training method is used. In this study, they are integrated into the training system. Some web pages are created to give the links to the description of the disease, the treatments and the PRP procedure with 2D anatomical images. In addition, the web pages also include the instruction of how to use the system and how the system is created.

3.5 Discussion

In this initial study, a Web-based VR training system in the field of neurosurgery, simulating the percutaneous rhizotomy procedure, has been introduced. A client-side architecture has been proposed in such a way that VRML and Java can be jointly used to provide both the simulation and presentation. Collision detection between the instrument and patient is critical to this simulation. We provide this by a pair of linked views: one from the eye of the surgeon, one from the eye of the needle. The needle-eye view uses VRML's built-in facility to test for collision between viewpoint and objects, and the collision information is passed to the surgeon's eye view via the Java EAI. The various parts of the procedure – marking anatomical landmarks, orienting the needle and inserting the needle up to point of collision – are all simulated by the application. The assessment element is integrated into the system. The simulation is deliberately kept simple in order to run efficiently on modest equipment.

During the development of the system, we have dealt with many changes and bugs in VRML viewers in order to make the system robust and usable in a wide range of situations. Different types and versions of the VRML plug-in also cause problems if we need to make the simulator available for different viewers. For example, the Craniotomy system (see Figure 3.16) in which we have been involved in research and implementation, was only developed with the WorldView VRML viewer. It had been implemented in a different way to make it available in other viewers. Eventually, only two popular browsers are focused: CosmoPlayer and Cortona. The source codes have been continuously updated to cope with the changes in the two viewers. Furthermore, different types and versions of Web browsers, such as Netscape and Internet Explorer, can cause problems. For example, Netscape 6 doesn't support Java for the plug-ins which were created for Netscape 4.x.

The simulator is available on a public access web site, and is therefore accessible to anyone with a PC, Internet connection and Web browser with VRML plug-in.

Chapter 4 Soft Object Modelling: SurfaceChainMail

4.1 Introduction

In the initial study described in the previous chapter, the interaction between rigid objects was considered. However, in many applications the objects involved are 'soft' and deformable when forces are applied. Thus, there is a need to study techniques for modelling the interaction involving rigid and soft objects in order to address a wider class of applications.

While there has been significant progress in the development of interactive simulation for rigid objects in the computer graphics community, there remains substantial work to be undertaken for soft objects. When a soft object is involved in an interaction, deformable modelling techniques are required to simulate the behaviour of the object during interaction. The speed of interaction is critical in applications such as surgical training. In order to accurately represent the structure of soft objects, many applications require large datasets to describe the structure realistically. For example, in medical applications, realistic 3D data is mostly generated by CT or MRI scanning. 3D objects are constructed from the 3D data with different resolutions. When an object contains a great detail of structure, a high resolution is required to show the details of the object. In addition, many soft objects result in large deformations when forces are applied. Furthermore, when the applied forces exceed a threshold (i.e. cutting) or the object is put together from smaller parts (i.e. suturing), it will result in topology changes of the object. Therefore, this study focuses on the development of fast deformable modelling techniques, with the capabilities of handling large deformations for large datasets and modifying the topology of objects in real time.

As described in the review section, the most commonly used methods in physically-based deformable modelling are FEM and mass-spring systems. Because of the simplifications and assumptions made in order to make them computationally feasible, these techniques are inadequate for the large deformations and for the large size of object. 3D ChainMail (Gibson, 1997), which is called Gibson ChainMail in this thesis, is a simple technique but capable of handling in real-time for large datasets and large deformations. In addition, changes in topology are easily modelled with this algorithm. However, the Gibson ChainMail was designed for volumetric objects. It is inapplicable to surface objects. In this study, a rather different approach, based on the Gibson ChainMail, is proposed, which can be applied to objects defined in a surface representation.

In section 4.2, the Gibson ChainMail technique is described and its limitations are discussed. Section 4.3 describes a generalisation of the Gibson ChainMail for non-uniform meshes. Section 4.4 shows how a 2D Gibson ChainMail mesh can be mapped to a surface in 3D, and used as a basis for deformable modelling. Section 4.5 describes the Web implementation and the surgical application including the techniques used in the system. Section 4.6 summarises and discusses the limitations of the work, and the motivation of further work.

4.2 The Gibson ChainMail Algorithm

In the Gibson ChainMail Algorithm, an object is modelled as a set of point elements, linked in a uniform rectilinear mesh. In contrast to FEM, where complex calculation is carried out on a (relatively) small number of mesh elements, the Gibson ChainMail carries out simple calculations on a (potentially) large number of elements. With the linked data structure, the interactivity for objects with high geometric complexity is achieved by using low-cost mathematical modelling which is performed locally on the elements, and by propagating the displacement from one element to the other via the links between the object elements.

4.2.1 Data Structure

The Gibson ChainMail algorithm requires each element to lie on a grid with fixed neighbours. For example, in 2D Gibson ChainMail, each element normally has four neighbours: left, right, top and bottom. In the 3D case, two additional neighbours, front and back neighbours, are linked to it. The data structures used in the Gibson ChainMail algorithm consist of the object data structure, such as the size and material properties of the object; the object element data structures, such as colour, position vector and connections to the neighbouring elements; and the data structure used to keep track of previous positions of moved elements. How much information is contained in the data structure is flexible. It depends on the application. For example, if the visual properties, such as opacity, are important, these features can then be stored for each element.

4.2.2 ChainMail process

Under the data structure just defined, the Gibson ChainMail algorithm carries out a deformation in two steps: a ChainMail step in which one element is moved, and the change is propagated to neighbouring elements; and a subsequent relaxation step, in which each element position is locally adjusted so that the system reaches a minimum energy state.

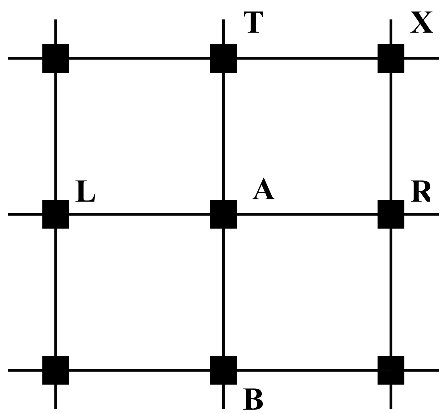


Figure 4.1: 2D ChainMail Arrangement : A is element to be moved

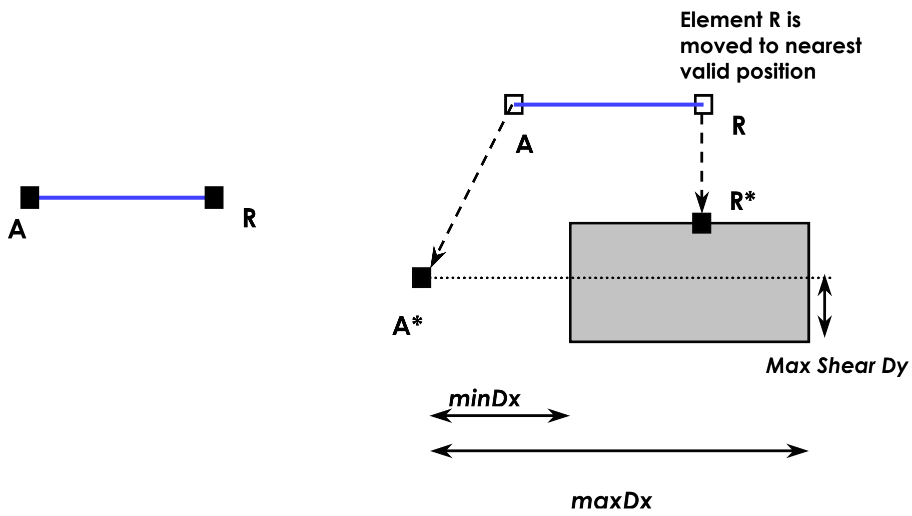
This can be illustrated more simply in 2D as shown in Figure 4.1. The elements are connected in a regular and rectilinear mesh arrangement, so that each element has four neighbours (right, left, top, bottom). Suppose element A is moved to a new position; its four neighbours R, L, T and B are candidates for adjustment as a result of A's move – A is known as the *sponsoring* element for these moves. The elements R, L, T and B are added to right, left, top and bottom candidate lists respectively. The right list is considered first. Figure 4.2(a) shows the initial positions of A and R. Figure 4.2(b) shows the valid region for R based on the new position of A, A*: it constrains the compression (through *minDx*), the stretching (through *maxDx*) and the shearing (through *maxShearDy*). If R lies within the region, then there is no change made and we move on to the left candidate list. If R lies outside the region, it is moved to the nearest point, R*, within the region – and its three other neighbours are added to the candidate lists. The position of element R is updated as follows:

$$\text{if } (x_R - x_A) < MinDx,\ x_R = x_A + MinDx;$$
$$\text{else if } (x_R - x_A) > MaxDx,\ x_R = x_A + MaxDx;$$

$$\text{if } (y_R - y_A) < -MaxShearDy,\ y_R = y_A - MaxShearDy;$$
$$\text{else if } (y_R - y_A) > MaxShearDy,\ y_R = y_A + MaxShearDy$$

The right candidate list continues to be processed in this way, with R being the sponsoring element for its right neighbour. Once the right list is empty, the left list, the top list and finally the bottom list are processed in turn. The left candidate list is processed in a similar manner, but with the left candidate being the sponsoring element for its left neighbour and the movement of a left element in the list results its three neighbours, left, top and bottom, to be added to their respective candidate lists. The top (or bottom) candidate list is also processed in a similar manner, but with the top (or bottom) candidate being the sponsoring element for its top (or bottom) neighbour and the movement of a top (or bottom) element in the list results only its top (or bottom) neighbour to be added to its respective candidate list.

Different settings for the parameters *minDx*, *maxDx*, *maxShearDy* (and corresponding *minDy*, *maxDy*, *maxShearDx* for vertical links) control the perceived softness of the material.



(a) (b)

Figure 4.2: (a) The original configuration of elements A and R; (b) Following move of A to A, the element R is moved to the nearest point of the valid region shown in grey*

A key to the success of the algorithm is that each element is processed at most once. Gibson (Gibson, 1997a; Gibson, 1997b) showed that while processing the top candidate list, any adjustments did not add new elements to the already completed right list. Suppose R and T satisfy the constraints with respect to A, and suppose X in the top list is adjusted if needed to satisfy constraints with respect to R as sponsoring element. Gibson showed that X lies within the valid region with respect to T as sponsoring element, for at least one valid position of T relative to A. This result was sufficient for Gibson to argue that X does not need to be added to the right list with T as sponsor.

This algorithm is, however, inappropriate for inhomogeneous materials. An extension to the algorithm for inhomogeneous materials (Schill et al., 1998) was achieved by sorting each candidate list by the size of the constraint violation at each step, where the element with the largest constraint violation was processed first. This enhanced ChainMail algorithm demonstrated its feasibility in modelling of inhomogeneous materials. However, the performance was significantly slower than the Gibson ChainMail.

4.2.3 Relaxation Process

The ChainMail propagation is then followed by the relaxation step, in which each element is locally re-positioned so as to be equidistant from its neighbours. Care should be taken in achieving this. As Gibson pointed out, if the element is moved to a new position which is the centroid of its neighbours, this will cause the boundary elements to move towards the centre and the object to shrink (see

Frisken-Gibson, 1999). Therefore the new position is chosen as follows: each neighbour is brought towards the element by an amount equal to an optimal link length, and then the centroid is calculated. This reduces excessive shrinkage of the boundary, and is equivalent to using the centroid for an interior element. This iterates for several times towards a minimum energy state of the whole system. The extension of the algorithm to 3D is straightforward: there are now six neighbours rather than four, but the concept is unchanged.

4.2.4 Discussion

The Gibson algorithm has been used successfully to model soft objects with large numbers of elements, and timing experiments reported in Gibson (1999) confirm that the computational cost scales linearly with the number of elements. However, there are a number of restrictions with the algorithm that require further development. The algorithm has been derived on the assumption of a volume graphics approach: objects are modelled on a voxel basis with links between each voxel. Thus, the elements are laid out in a regular, rectilinear mesh. In this study, the Gibson algorithm is extended to surface-based representation of objects as described in the following sections. To help in understanding this work, section 4.3 first describes how the original algorithm can be extended from regular to irregular rectilinear meshes.

4.3 Extension to Non-uniform Grids

The Gibson ChainMail algorithm is defined only for uniform meshes. However it is straightforward to handle non-uniform meshes by replacing the absolute constraints of Figure 4.2 by constraints expressed relative to the separation of the mesh lines. This generalisation to non-uniform meshes will enable us to use the ChainMail technique in a wider class of modelling applications. By non-uniform, it means any mesh topologically equivalent to a uniform rectilinear mesh – that is, a mesh with connections from each interior element to left, right, top, bottom, front and back elements. The mesh can be rectilinear or curvilinear. The technique for curvilinear mesh will be explained in the next Chapter.

The constant geometric constraints are inappropriate for non-uniform grids. These constraints can determine the softness of objects only when the objects are sampled with uniform grids. Otherwise, it produces inconsistent results. The proposed modification is to separate each geometric constraint into two parts: first, a material parameter which controls the softness, and secondly, the distance between the linked elements. Thus, the entire geometric constraints will vary when the length of the links is changed.

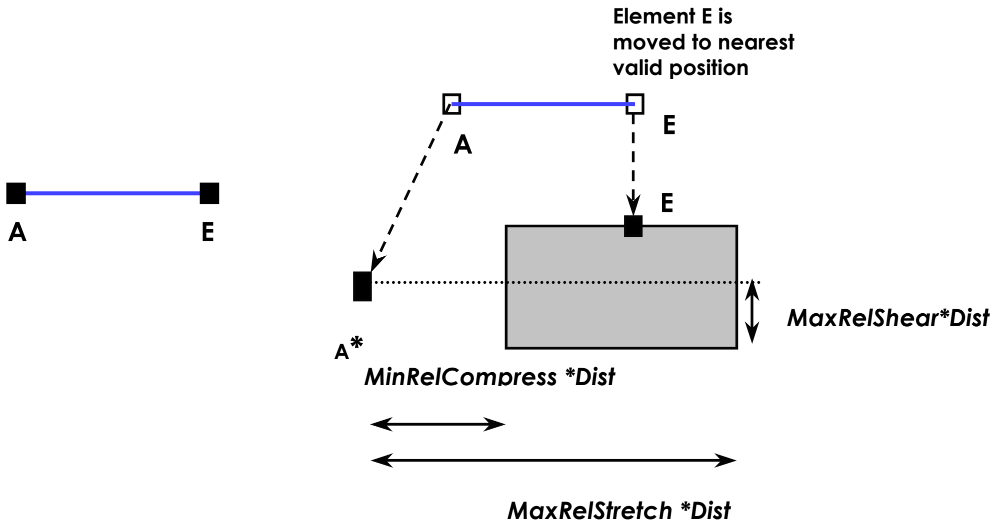


Figure 4.3: The constraints between element A and its neighbour E

Again suppose A has been moved to a new position, and its right neighbour, E, has become a candidate for movement in the chain reaction. Suppose *Dist* is the distance between A and E. The valid region for E is shown in Figure 4.3, and is expressed by material parameters, *MaxRelStretch*, *MinRelCompress* and *MaxRelShear*, relative to the separation, *Dist*. With this modification, we can apply ChainMail to non-uniform grids. By tuning the values of the material parameters, objects with different elasticity can be modelled in an efficient way.

4.4 Extending to Surfaces in 3D: SurfaceChainMail

As analysed earlier, the Gibson ChainMail algorithm was designed for 3D volume modelling where the elements are on a regular, rectilinear mesh – the usual voxel-type structure. Many objects, however, are perfectly well modelled as surfaces. This section describes a new variation of ChainMail, so called SurfaceChainMail, which is designed specifically for surfaces. The deformation process is decomposed into two steps. First, a deformation in the plane of the surface is carried out, using the extension to non-uniform grids just described. Secondly, a deformation in the direction of the surface normal is performed. The output result is the combination of these two deformation processes.

The data structure defined in the Gibson ChainMail is extended to separate the material properties into two parts: the material parameters and the distances between the links. The distance between the links is stored in the data structure explicitly in order to reduce run time calculation, as it is repeatedly used many times during the deformation process.

Before performing the deformation process, the forces are decomposed into two components: the component within the tangent plane and the component along the 3D surface normal. Each component causes the movement in different direction.

4.4.1 Deformation in Tangent Plane

In the first process, a 2D mesh is mapped onto the surface of the 3D object (as with texture mapping in graphics). Thus a 2D mesh, S, is defined in the tangent plane which is described with parameters (u,v). The force at a point is separated into components parallel to the surface and normal to the surface; the tangent plane component is expressed in the (u,v) coordinate system. If the 2D mesh is uniform, the Gibson's ChainMail technique is performed on it, otherwise, the 2D ChainMail technique with relative constraints (as described in section 4.3) is performed on the 2D mesh. The updated positions in the (u,v) space are then transformed back to the surface of the 3D object in (x,y,z) space.

Sometimes it is difficult to map the rectilinear 2D mesh, S(u,v), onto the surface of the 3D object directly. In this case, it is necessary to transform to an intermediate surface, such as a cylinder, which maps more easily to the targeted 3D surface. Then a transformation is performed from the intermediate surface to the target 3D surface. Clearly, this intermediate surface mapping process will involve overheads during the interaction.

4.4.2 Deformation Normal to Surface

In the second part of the process, deformation normal to the surface is considered. The surface normal at an element is obtained by the local average of the normals of surrounding facets. If the displacement of a moved element in its normal direction is less than *MaxRelShear*Dist*, the shearing constraint, then the position of a neighbour of the moved element is unchanged during the ChainMail processing; but if the displacement is greater than this, the neighbour is moved in its normal direction so that the separation in that direction between the two elements is *MaxRelShear*Dist*. The deformation process in the direction of the surface normal for each element is determined by the following statements:

if $(n - n_{neighbour} > MaxRelShear*Dist)$, *then* $n = n_{neighbour} + MaxRelShear*Dist$;

else if $(n - n_{neighbour} < - MaxRelShear*Dist)$, *then* $n = n_{neighbour} - MaxRelShear*Dist$;

where n is the displacement of the element in its surface normal direction, $n_{neighbour}$ is the displacement of its neighbour in its surface normal direction (i.e. the normal at the neighbour position) and *MaxRelShear*Dist* is the shearing constraint.

4.4.3 Combination of Deformation Processes

The results from these two processes are combined together to produce a deformed new shape of the 3D surface. Figure 4.4 outlines the whole deformation process: Two deformation processes are carried out in parallel, each process uses different algorithm but the same principle to update the position of the elements in different directions. The two processes merge together to produce a final result.

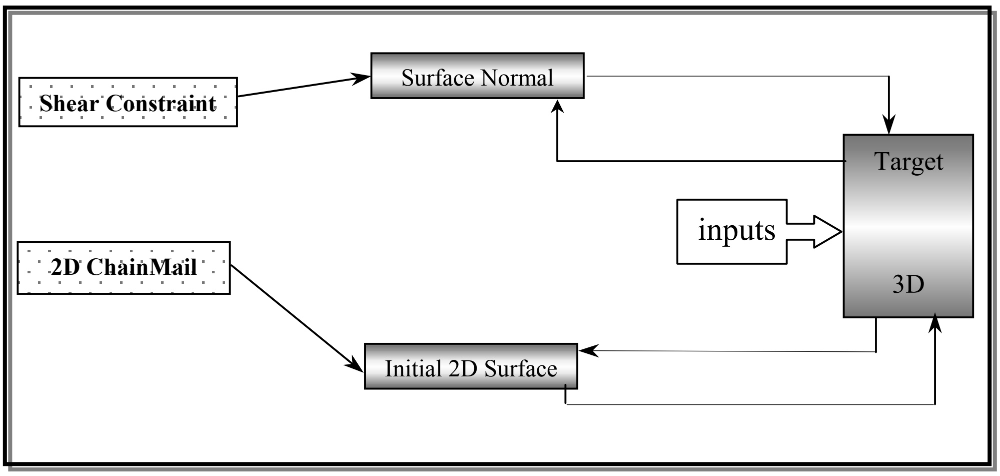


Figure 4.4: SurfaceChainMail process.

If an intermediate surface is needed, the deformation process is summarized in Figure 4.5.

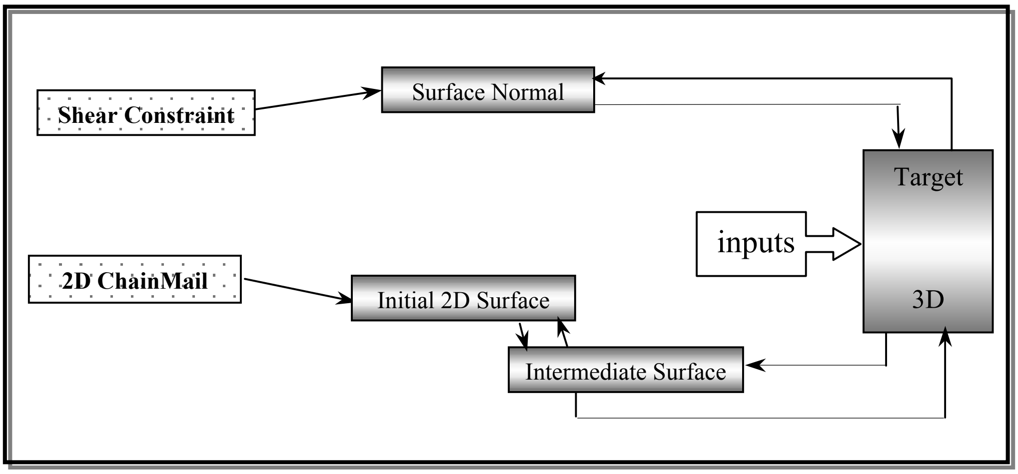


Figure 4.5: SurfaceChainMail process with intermediate surface.

Figure 4.6 shows the examples of using these techniques. In the three pictures to the left, a cut cylinder is pulled and pushed, where the 2D ChainMail is applied directly to the initial 2D surface; to the right, a cut sphere is deformed.

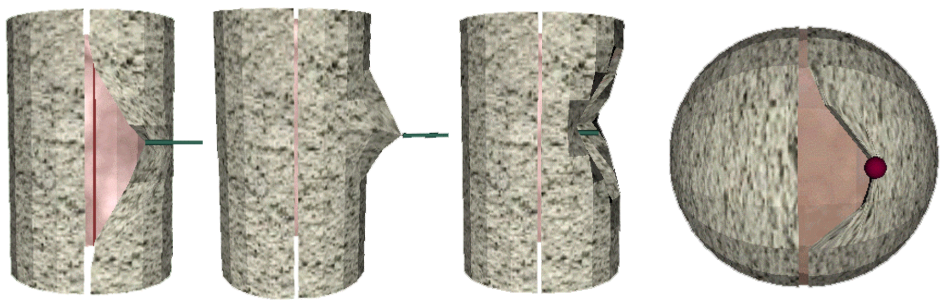


Figure 4.6: Examples of deformable surface using SurfaceChainMail

4.5 Using SurfaceChainMail for Surgical Applications

One of the primary goals for developing the fast algorithm for deformation of surface based objects in this research is to facilitate web-based surgical training for a wide class of applications. Thus, SurfaceChainMail has been implemented as a Web-based application so that it can be used as a surgical training simulator in the collection of such tools being developed (Web-Based Surgical Simulators, 2002). Again, this system employs a client-based solution, where the data and applications are located on the web server.

4.5.1 System Overview

The system uses a combination of VRML and Java, as in the PRP simulator: VRML to provide the visual display and Java code to provide the real-time soft tissue modelling using SurfaceChainMail. The Java External Authoring Interface provides the link between the two. The system structure is shown in Figure 4.7. The cutting process is available when the cutting simulation is performed.

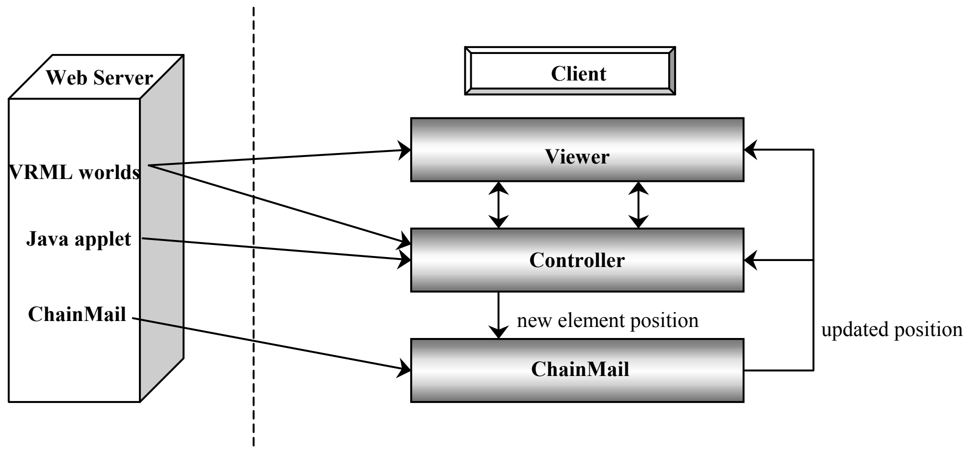


Figure 4.7: System architecture in the cutting process.

The system has three main components: the *Viewer*, the *Controller* and the ChainMail engine. When the simulator Web page is loaded from the server, two VRML worlds are downloaded into the *Viewer* and the *Controller* respectively, a Java applet is downloaded into the *Controller*, and the mesh structure is downloaded into the ChainMail engine. As a Java applet cannot read data from VRML files directly, the mesh structure is placed on the web server, and is used to build the linked data structure for the SurfaceChainMail process. When a user alters a position of an element from the controller component, this is transmitted to the ChainMail engine, which then computes the resulting new mesh positions. These are transmitted back to the *Viewer* and *Controller* components. The Java EAI for VRML provides the 'glue' which allows the components to communicate with each other over the Internet.

4.5.2 Surgical Training Application

One of the advantages of the Gibson ChainMail approach is that it supports topological changes during the interactive modelling: links between an element and its neighbours can be easily disconnected when needed. This applies in exactly the same way to SurfaceChainMail. This has allowed us to build a general simulator for the cutting and separation of layers of soft tissue. It models a very fundamental technique in surgery of all types for dissecting out structures as part of a more complex operation. In neurosurgery a common start to many operations involves the dissection of the covering layers of the brain (- the meninges) to allow access to deeper structures such as blood vessels.

Interface

Figure 4.8 shows a screenshot of the simulator. There are two linked VRML worlds: the trainee manipulates the cutting instrument in the lower left browser window – the *Controller*, while a view of the resulting deformation is provided for the trainee, and the observers, in the upper window – the *Viewer*. This interface design is similar to the PRP simulator, which is described in Chapter 3, in the way that the controls are concentrated in one window for ease of use. Instead of using widgets to manipulate the instruments, in this design, we are able to rotate or insert the instrument directly on the virtual instruments in the *Controller* through the mouse. This improves the realistic visual effects compared to the interface in the PRP simulator.

In order to manipulate the instrument easily, some strategies are applied. For example, when forceps are chosen from the toolbox in the *Controller*, the viewpoint is automatically moved to the best view from where the pulling or pushing of soft tissue can be performed. Otherwise, it will be jumped to the best view from where the cutting of soft tissue can be performed. Thus, the user does not need to manipulate the viewpoint for each task. In addition, the browser of the *Controller* is "frozen" so that the user cannot manipulate this viewpoint. Consequently, the dragging action can be easily made in this way.

Thus, the procedure of performing the cutting task is as follows. The user loads the application onto the local machine from the web server. Once the Java applet is loaded onto the client side, the user can start to perform the surgical tasks. Firstly, they choose a surgical tool, such as scissors and forceps, from the *Controller*. If forceps are chosen, they can pull or push the corner of the tissue. On the other hand, if scissors are chosen, they can start to cut the tissue. If further cutting process is needed, they can pull or push the tissue further with the forceps, and then cut again with the scissors. This procedure can be repeated until a satisfactory opening is made.

Cutting

Performing cutting is a difficult challenge. Deformation and cutting are not independent processes. They often occur simultaneously. As described earlier, many strategies have been suggested for performing realistic deformation. One of the commonly used methods is to use a pre-processing phase, which implies that real time topology change is impossible. Therefore, Cotin et al. (1999) introduced a hybrid approach which applies the accurate model with a pre-processing scheme on the area where cutting is not performed, and a model which does not require a pre-processing strategy on the area where cutting may be performed. This approach is very efficient but with the drawback of specifying which area of the object may be cut in advance. The recent work by Ganovelli et al. (2000) described a multi-resolution approach to the mass-spring method, which optimises performance by representing only the critical parts of the object at high resolution by degrading data resolution smoothly with the increase of distance from the parts. The modification of topology is made by first clipping the tetrahedra that are interacted with a scalpel, and then updating the multi-resolution triangulation to the new topology. This approach allows real time topology change for a certain number of tetrahedra.

In this simulation, the linked data structure in SurfaceChainMail allows the interactive modification of topology by connecting or disconnecting the links between the nodes of the object. This simple scheme can be applied in time-critical applications. This system demonstrates the feasibility that the SurfaceChainMail can support topology change – specifically cutting of the soft tissue.

To decide a cutting path, the first step is to check whether there is a collision between the instrument and the soft tissue. As indicated earlier, VRML does not support collision detection between objects in the scene. The method of collision detection used for the PRP simulator is not appropriate for this application, as it does not specify which pairs of polygons are colliding. This is useful information for the cutting simulation. The contact points can be used to decide whether a cut is taken place according to the threshold of the forces or displacements applied on the point.

In this simulation, cutting is simulated when the user defines a ‘cut path’ using the mouse as a virtual instrument. When the virtual scissors are selected from the toolbox, the mouse acts as the role of virtual scissors. To determine whether the virtual scissors are in contact with the soft tissue, a *TouchSensor* node is placed on the tissue surface. So when the mouse “touches” the tissue, it will specify where the contact points are. Elements close to the cut path are identified by a searching process and any links with their neighbours which are crossed by the cut path are removed from the data structure. The cutting procedure is described as follows:

A small volume of cube is defined according to the contact information from the *TouchSensor* node. If there are elements within the cube, then it is necessary to check all the links between the elements and their neighbours to see whether they are intersected by the cut path. If they are, then the links from the object element data structure will be deleted.

Figure 4.8 shows surface based deformable modelling and cutting with web-based environment.

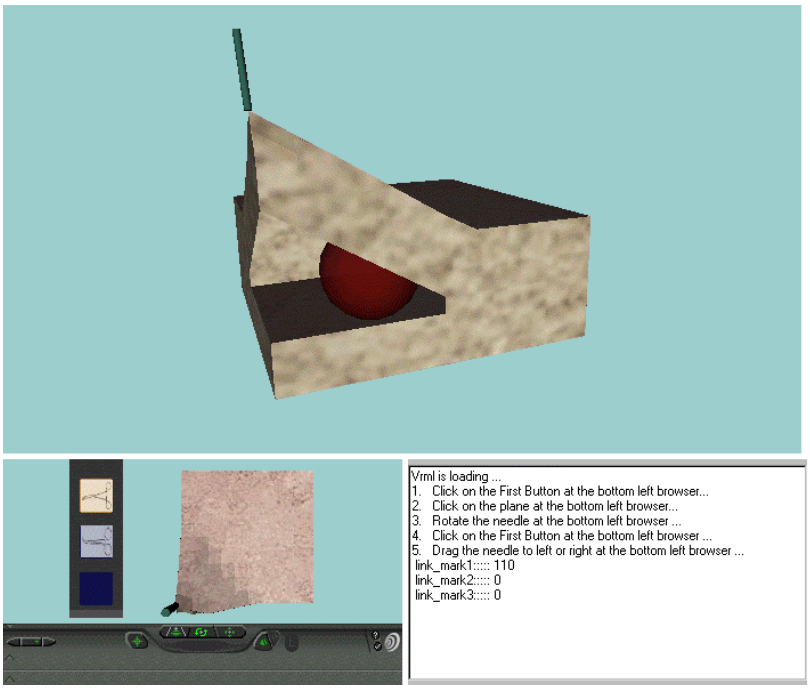


Figure 4.8: Surface based deformable modelling and cutting with web-based environment.

The browser window at the bottom left is the *Controller*, where the user manipulates the instrument and performs deformation and cutting process. The instrument can be selected at the left bar of the window and rotated as well as inserted. The window at the top displays the results from the *Controller*. The applet *Viewer* at the bottom right is the monitor.

This simple solution provides a very fast result to find the contact point and make a cut. The drawbacks of this technique are that, if the speed of the “virtual scissors” is too fast, the *TouchSensor* could miss sensing some contact points along the cut path and give improper results. Moreover, the strategy of removing links from the data structure introduces artefacts along the edge of the cutting path, giving a zig-zag effect. Possible solutions to this are to increase the resolution of the tissue, or use a better cutting algorithm by splitting a quadrilateral intersected by the cut path into several small quadrilaterals. However, these possibilities are not investigated in this prototype system.

Deformation is simulated when a virtual instrument pulls or pushes the object: an element is moved to a new position by the instrument and SurfaceChainMail calculates the resulting effect. In Figure 4.8 above, we see two layers of tissue that have been cut, and teased apart using the instrument. One side of the edge is fixed, so when the pushing or pulling action exerts at the corner of the tissue, only one side of the cutting path is deformed under the displacement.

4.6 Discussion

The SurfaceChainMail, a variation of the 3D ChainMail algorithm introduced by Gibson, has been developed for volumetric modelling. It uses non-uniform grids and can be used for modelling surfaces. Its simplicity makes it suitable for real-time deformable modelling in Web-based surgical simulators. It has been used to develop a training simulator for a simple cutting of soft tissue.

The present approach works well for objects where there is a simple mapping from the (u,v) mesh to the 3D object – such as the cylinders and spheres used in this study. However, this SurfaceChainMail approach has several limitations:

(1) The conversion from a 3D Cartesian space to its 2D parametric space and the inverse of the conversion produce more overheads in contrast to the original ChainMail algorithm, which means that it is slower than the original ChainMail algorithm.
(2) This quadrilateral surface based representation makes the accurate cutting process difficult to achieve. This is because the quadrilateral cannot be split into arbitrary meshes when it is intersected by the cutting path.
(3) Although the extension to non-uniform grids provides more flexibility over the original ChainMail, it is still restricted to quadrilateral meshes, which has limited flexibility in data representation.

Therefore, the work goes on to describe a further development in which the surface will be defined as a general unstructured mesh: this mesh is triangulated, and the sides of the triangles become the mesh links. This will extend to 3D tetrahedral meshes to give a chainmail-type method for volumetric modelling on unstructured meshes.

Chapter 5 Soft Object Modelling: Generalised ChainMail

5.1 Introduction

There have been two developments to the original ChainMail algorithm. In Schill et al (1998), the algorithm is re-arranged to allow the compression, stretching and shear constraints to vary with each element. This enables inhomogeneous volumes to be modelled, but at the cost of a significant increase in compute time (see Gibson, 1999). The SurfaceChainMail, described in last chapter, extends the algorithm to non-uniform rectilinear meshes, by replacing the absolute constraints of Figure 4.2 by constraints expressed relative to the mesh spacing. A key restriction of all the work to date has been the need to use a mesh which is rectilinear.

The research presented in this chapter is to extend the ChainMail concept to handle arbitrary meshes. Thus for 3D volumes, the object can be modelled as a set of elements connected in a tetrahedral mesh structure, allowing much greater flexibility in the arrangement of the elements than has been possible before. An important aspect of our work is that both surfaces and volumes can be handled by the same basic approach: a surface represented as a triangular mesh in 3D space is deformed using the same algorithm as a volume represented as a 3D tetrahedral mesh.

5.2 Generalised ChainMail

In this section, the extension of the ChainMail algorithm, which is called 'Generalised ChainMail' in this thesis, is described. Suppose that we have a mesh in which each element is arbitrarily positioned, and is linked to any number of neighbouring elements. The mesh can be of any dimension, and can be embedded in a space of the same or higher dimension. Thus we can have as particular cases:

- A 1-dimensional mesh of elements, as a 1D line, or as a curve on a 2D plane, or in a 3D space
- A 2-dimensional mesh of elements, for example a triangular mesh, lying in a 2D plane, or forming a surface in a 3D space
- A 3-dimensional mesh of elements, for example a tetrahedral mesh, in a 3D space

Indeed the original ChainMail arrangement of a rectilinear grid is another particular case.

This section begins by describing the data structure used to store the information such as positions of the elements and the links between them. How the movement of one element causes a constrained movement of a neighbour is then described, followed by an explanation of the entire cascade of element moves that occur when one element is moved. Crucially the algorithm has a similar termination property to the original ChainMail: after the movement of one element, the cascade involves re-positioning each element no more than once, and it is shown next. In order to ease the description, the algorithm for the 2D planar case, with a triangular mesh, will be described. However in the final part of this section, the comments on the changes required for other dimensionalities are given.

5.2.1 Data Structure

The data structure to store a Generalised ChainMail object is straightforward, and similar to that used in the original algorithm (see Gibson, 1999). An object is stored as:

Object = {list of elements, dimension of object, compression factor $(\alpha_{\min})$, stretch factor $(\alpha_{\max})$, shear factor (β) }

Each element is stored as:

Element = {

Identifier, *I*

Element value, (*colour, transparency*)

Element position, (*x, y*)

Processing state (*flag* = `updated | not_updated`)

Number of neighbours, *N*

List of neighbours, { (I_j), *j=1,2,..*},

where I_j is the identifier of the j^{th} neighbour of the element

}

5.2.2 Updating the Position of an Element

Suppose element A has been selected and moved to a new position. Suppose element B is one of its neighbours. As before, we say that A is the sponsoring element for B. Figure 5.1(a) shows the starting situation, before A is moved; Figure 5.1(b) shows the final situation, where following A's move to A*, B is moved to the nearest point, B*, of a valid region shown in grey.

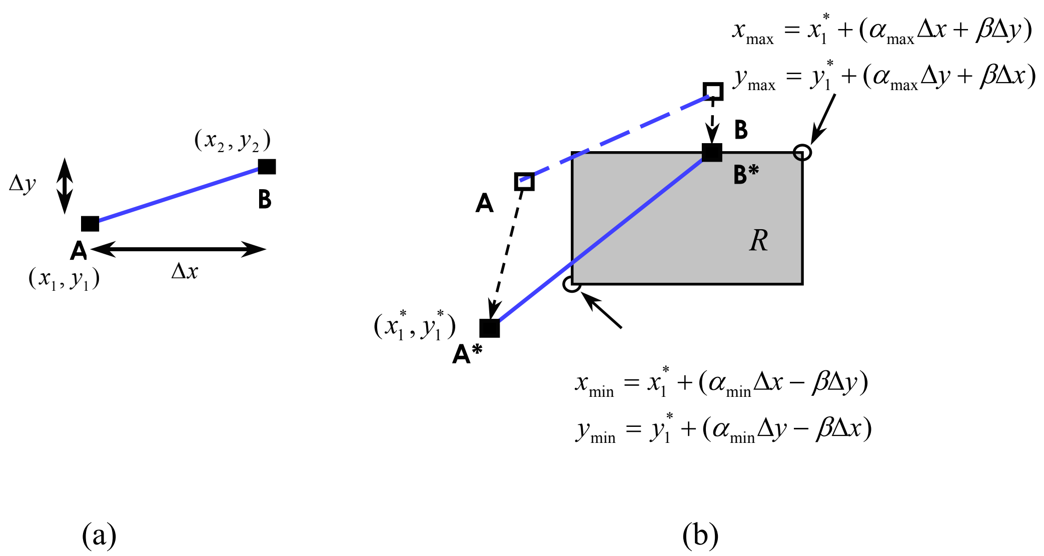


Figure 5.1: Generalised ChainMail - Updating Rule : (a) original position of A and B; (b) following move of A to A, the element B is moved to B*, the nearest point of the valid region shown in grey*

Specifically, suppose A is the point (x_1, y_1) and B is the point (x_2, y_2). We define

$$\Delta x = |x_2 - x_1|;\ \Delta y = |y_2 - y_1|$$

Two 'softness' parameters, $\alpha_{\min}$ and $\alpha_{\max}$, which control respectively the amount the object can be compressed, and stretched, are also defined. A further parameter, β, controls the shearing that is allowed. These together define a valid region, *R*, for element B, following the move of element A to its new position A*, say (x_1^*, y_1^*). This valid region is given by:

$$R = \{(x, y) : x_{\min} \le x \le x_{\max};\, y_{\min} \le y \le y_{\max}\}$$

where (supposing first that both $x_2 \ge x_1$ and $y_2 \ge y_1$ as in Figure 5.1(b)):

$$\begin{aligned} x_{\min} &= x_1^* + (\alpha_{\min}\Delta x - \beta\Delta y) \\ x_{\max} &= x_1^* + (\alpha_{\max}\Delta x + \beta\Delta y) \\ y_{\min} &= y_1^* + (\alpha_{\min}\Delta y - \beta\Delta x) \\ y_{\max} &= y_1^* + (\alpha_{\max}\Delta y + \beta\Delta x) \end{aligned}$$

Notice the significant difference from the original ChainMail algorithm, in that the softness and shearing parameters are expressed *relative* to the length of the original link between A and B, rather than as *absolute* distance values. This modified algorithm can be directly applied to curvilinear meshes.

If $x_2 < x_1$, or $y_2 < y_1$ the definition of *R* changes in a straightforward manner. For example, if $x_2 < x_1$, then the definitions of $x_{\min}$ and $x_{\max}$ change to:

$$x_{max} = x_1^* - (\alpha_{min}\Delta x - \beta\Delta y)$$
$$x_{min} = x_1^* - (\alpha_{max}\Delta x + \beta\Delta y)$$

5.2.3 Cascade of Element Moves

Now consider the whole sequence of element moves that occur when one element of the mesh is moved to a new position. Again a diagram is useful to help understand the process. In Figure 5.2, we see a mesh of nine elements, and suppose that element 1 is moved to a new position.

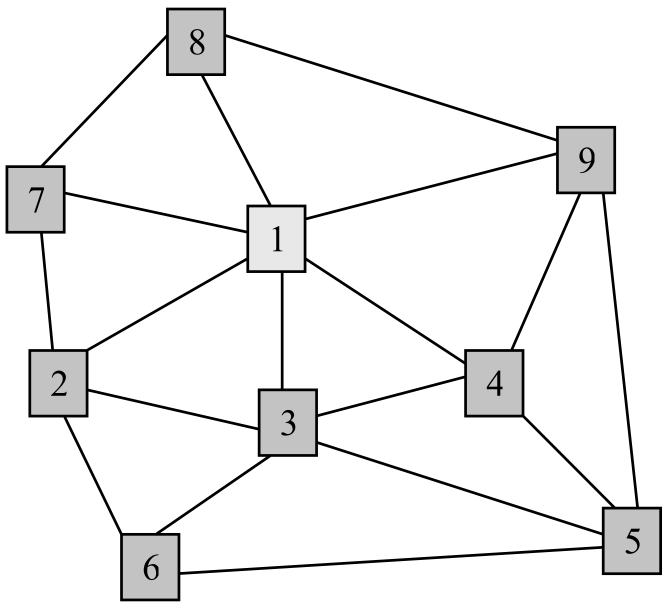


Figure 5.2: Linked elements: cascade effect of moving element 1

The algorithm proceeds as follows:

1. The (x, y) position of the moved element, element 1, is updated.
2. The neighbours of element 1 are added to a queue of elements that are candidates for moving; this is called the *waiting list*. At this stage, therefore, elements 2, 3, 4, 7, 8, 9 are added to the waiting list, with element 1 as their sponsoring element.
3. For each element in the waiting list, the following steps are carried out:

   3.1 Check the *processing_state* of the element

   3.2 If it is `updated`, it is removed from the waiting list and the next element in the list is continued.

   3.3 If it is `not_updated`, its position is checked against the valid region *R* with respect to its sponsoring element as described in section 5.2.2. If it lies outside *R*, then it is moved to the nearest point of *R* as described in section 5.2.2, and its neighbouring elements are added to the front of the waiting list, with this element as sponsor. Finally the *processing_state* of the element is changed to `updated` to indicate its position has been updated.

Thus, following through step 3 of the algorithm for the example, element 2 will be the first on the waiting list to be processed. It has not yet been updated, and so its position will be checked against the valid region *R* and moved if necessary. If it is indeed moved, its neighbouring elements (3, 6, 7, 1) are added to the waiting list, with element 2 as sponsor. The *processing_state* of element 2 is changed to indicate it has been updated. The algorithm continues, elements 3, 6 and 7 being processed in the same way as element 2, but with a different sponsor; element 1 is discarded because its *processing_state* will indicate it has already been updated.

The algorithm continues until the waiting list is exhausted.

We would like the deformation to occur in as smooth a fashion as possible. In particular, we would like the cascade effect of deformations to spread out uniformly from the initial element – rather than to deform first in one principal direction, then another and so on. This motivates a modification to the basic algorithm above, where we try to update elements in as uniform a fashion as possible. The idea is illustrated in Figure 5.3. The waiting list is created in a tree arrangement as shown. The basic algorithm creates a tree of increasing depth, with the result that the branch from the first neighbour of the initial element is processed to its outer leaves, before the second neighbour is processed. A better strategy that has been found in practice is to pause the processing of the first branch at a certain depth of the tree (here at level 2), and move to processing the second branch. Again this is limited to a certain depth, whereupon it is paused and the third branch considered; and so on. The first branch resumes when the last branch is halted.

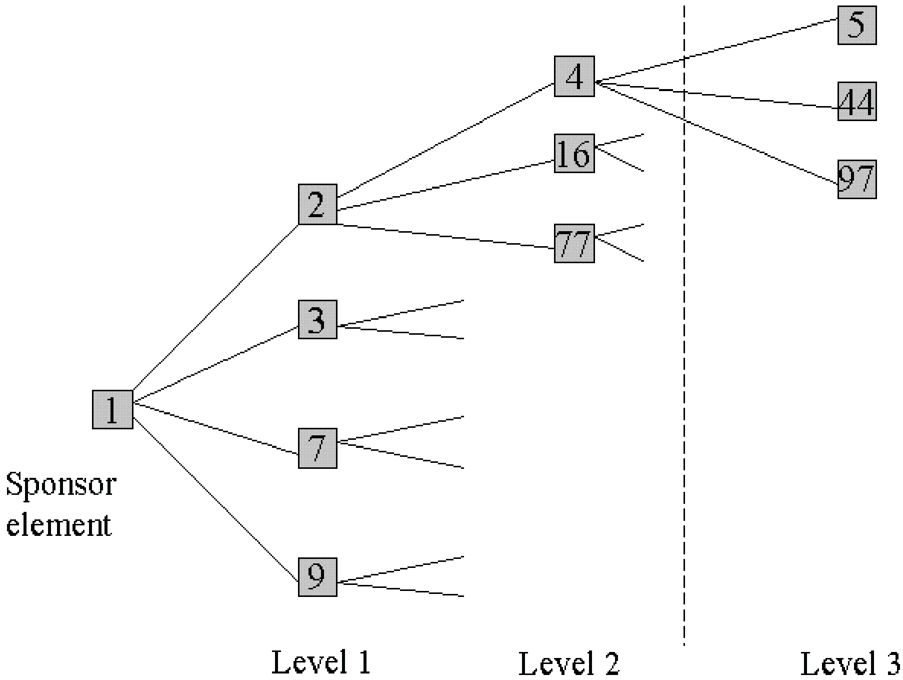


Figure 5.3: Processing the Waiting List

This gives a much improved visual effect. Indeed as mentioned in section 4.2, it has been noticed that the original ChainMail algorithm suffers a similar problem: the right list is processed to its end, before the top list is considered. When the material is very soft, especially when the shear constraint is relatively large, there is a ripple effect as the deformation completes its move to the right before moving upwards.

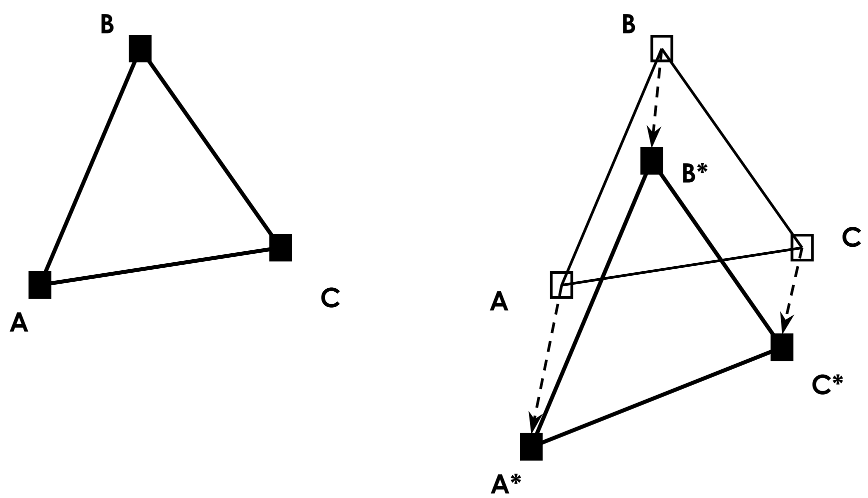
5.2.4 Termination Property

In this section, the action of processing each element no more than once is justified. First we explain the problem. Consider the simple network shown in Figure 5.4. There are three elements: A, B and C, at positions

$(x_1, y_1), (x_2, y_2)$ and (x_3, y_3). Suppose A is moved to a new position, A*. With A as sponsoring element, the algorithm first moves B to new position, B*; it then moves C to C*. The issue of concern is: having moved B and C independently to satisfy the constraint with respect to A, do we need to revisit their positions with respect to each other. If we had to do this, the algorithm could potentially cycle indefinitely. Fortunately it will be shown that, after the moves to B* and C*, C* satisfies constraints with respect to the movement of B to B*. It will be assumed:

$$x_3 \geq x_2 \geq x_1 \quad \text{and} \quad y_2 \geq y_3 \geq y_1$$

- the other cases can be proved in a very similar way.



(a) (b)

Figure 5.4: (a) The original positions of A, B and C; (b) Following A's move to A, B is moved to B* and C to C**

As noted that B* lies within the rectangle R_{21} (valid region for B with respect to A) given by:

$$R_{21} = \{(x, y) : x_{21,\min} \le x \le x_{21,\max};\; y_{21,\min} \le y \le y_{21,\max}\}$$

where

$$\begin{aligned}
x_{21,\min} &= x_1^* + (\alpha_{\min}\Delta x_{21} - \beta\Delta y_{21}) \\
x_{21,\max} &= x_1^* + (\alpha_{\max}\Delta x_{21} + \beta\Delta y_{21}) \\
y_{21,\min} &= y_1^* + (\alpha_{\min}\Delta y_{21} - \beta\Delta x_{21}) \\
y_{21,\max} &= y_1^* + (\alpha_{\max}\Delta y_{21} + \beta\Delta x_{21})
\end{aligned}$$

with $\Delta x_{21} = |x_2 - x_1|$ and $\Delta y_{21} = |y_2 - y_1|$. Likewise it is known that C* lies within the rectangle R_{31}, the valid region for C with respect to A:

$$R_{31} = \{(x, y) : x_{31,\min} \le x \le x_{31,\max}; y_{31,\min} \le y \le y_{31,\max}\}$$

where

$$\begin{aligned}
x_{31,\min} &= x_1^* + (\alpha_{\min}\Delta x_{31} - \beta\Delta y_{31}) \\
x_{31,\max} &= x_1^* + (\alpha_{\max}\Delta x_{31} + \beta\Delta y_{31}) \\
y_{31,\min} &= y_1^* + (\alpha_{\min}\Delta y_{31} - \beta\Delta x_{31}) \\
y_{31,\max} &= y_1^* + (\alpha_{\max}\Delta y_{31} + \beta\Delta x_{31})
\end{aligned}$$

with $\Delta x_{31} = |x_3 - x_1|$ and $\Delta y_{31} = |y_3 - y_1|$.

Now focus on B and C. Suppose B has been moved to B*; if C is moved to C*, does it lie within the valid region with respect to the movement of B to B*? That is, does C* lie within the rectangle R_{32}, given by:

$$R_{32} = \left\{(x,y) : x_{32,\min} \le x \le x_{32,\max};\ y_{32,\min} \le y \le y_{32,\max}\right\}$$

where

$$\begin{aligned} x_{32,\min} &= x_2^* + (\alpha_{\min}\Delta x_{32} - \beta\Delta y_{32}) \\ x_{32,\max} &= x_2^* + (\alpha_{\max}\Delta x_{32} + \beta\Delta y_{32}) \\ y_{32,\min} &= y_2^* - (\alpha_{\max}\Delta y_{32} + \beta\Delta x_{32}) \\ y_{32,\max} &= y_2^* - (\alpha_{\min}\Delta y_{32} - \beta\Delta x_{32}) \end{aligned}$$

with $\Delta x_{32} = |x_3 - x_2|$ and $\Delta y_{32} = |y_3 - y_2|$? The location of B* is unknown, just that it lies within the region R_{21}. Thus, in exactly the same way as Gibson in (Gibson, 1997), we seek to answer a weaker question: does C* lie within the rectangle $\overline{R_{32}}$ defined as:

$$\overline{R_{32}} = \left\{(x,y) : \overline{x_{32,\min}} \le x \le \overline{x_{32,\max}};\ \overline{y_{32,\min}} \le y \le \overline{y_{32,\max}}\right\}$$

where

$$\begin{aligned} \overline{x_{32,\min}} &= x_{21,\min} + (\alpha_{\min}\Delta x_{32} - \beta\Delta y_{32}) \\ \overline{x_{32,\max}} &= x_{21,\max} + (\alpha_{\max}\Delta x_{32} + \beta\Delta y_{32}) \\ \overline{y_{32,\min}} &= y_{21,\min} - (\alpha_{\max}\Delta y_{32} + \beta\Delta x_{32}) \\ \overline{y_{32,\max}} &= y_{21,\max} - (\alpha_{\min}\Delta y_{32} - \beta\Delta x_{32}) \end{aligned} \quad ?$$

That is, the co-ordinates of B* being replaced with limits of the rectangle R_{21} within which we know it lies.

So we need to show, for the *x*-coordinate of C*:

$$x_3^* \in \left[\overline{x_{32,\min}}, \overline{x_{32,\max}}\right]$$

First consider the lower limit. We have:

$$x_3^* \geq x_{31,\min} = x_1^* + (\alpha_{\min}\Delta x_{31} - \beta\Delta y_{31})$$

But

$$x_1^* = x_{21,\min} - (\alpha_{\min}\Delta x_{21} - \beta\Delta y_{21})$$

Hence

$$x_3^* \geq x_{21,\min} + \alpha_{\min}(\Delta x_{31} - \Delta x_{21}) - \beta(\Delta y_{31} - \Delta y_{21})$$

Since $x_3 \geq x_1$ and $x_2 \geq x_1$, and since $x_3 \geq x_2$, we have:

$$\Delta x_{31} - \Delta x_{21} = |x_3 - x_1| - |x_2 - x_1| = |x_3 - x_2| = \Delta x_{32}$$

Also,

$$\Delta y_{31} - \Delta y_{21} = |y_3 - y_1| - |y_2 - y_1| \leq |y_3 - y_2|$$

Hence finally

$$x_3^* \geq x_{21,\min} + (\alpha_{\min}\Delta x_{32} - \beta\Delta y_{32}) = \overline{x_{32,\min}}$$

as required. The proof for the upper limit follows in a similar way, and likewise the proof for the *y*-coordinate of C*.

Thus it has been shown that C* lies within the rectangle $\overline{R_{32}}$, and we are justified therefore in not re-visiting the link between B and C.

5.2.5 Relaxation Step

In the ChainMail process, the deformation is governed by purely geometric constraints. This raw shape is then refined in a relaxation process aimed to minimize the system energy. This is achieved by locally adjusting the position of each element.

Suppose an element *P* has three neighbours: $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$. Figure 5.5a shows the arrangement before deformation. The point *P* is at distance δ_i from P_i. After deformation, we suppose that *P* has moved to P^*, and likewise P_1 to P_1^* etc, as shown in Figure 5.5b. The elastic potential energy of the system locally at P^* is defined as proportional to the square of the additional displacements of P^* from P_i^*, compared with the original displacements δ_i. Thus, in Figure 5.5b, one needs to minimize the square of the displacements of P^* from

Q_1, Q_2, Q_3. This is achieved by choosing the relaxed position of P^* to be the centroid of the triangle $Q_1Q_2Q_3$. Thus, in detail, suppose:

$$\delta_1 = P - P_1; \quad \delta_2 = P - P_2; \quad \delta_3 = P - P_3$$

and define:

$$Q_1 = P_1^* - \delta_1; \; Q_2 = P_2^* - \delta_2; \; Q_3 = P_3^* - \delta_3$$

We want to minimize:

Energy term, E depends on:

$$\sum_1^3 \left\{ \left(\left[P^* \right]_x - [Q_i]_x \right)^2 + \left(\left[P^* \right]_y - [Q_i]_y \right)^2 \right\}$$

where $\left[P^* \right]_x$ for example is the x-coordinate of P^*. Equating the partial derivatives $\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}$ to zero, gives the optimal relaxed position for P^* as the centroid of $Q_1Q_2Q_3$. That is:

$$P_{relaxed} = \frac{Q_1 + Q_2 + Q_3}{3}$$

Note that the strategy of placing P^* at the centroid of $Q_1Q_2Q_3$, rather than $P_1^*P_2^*P_3^*$, is analogous to Gibson's strategy for the rectangular grid case (Gibson, 1999), where she adjusts points before calculating the centroid in order to avoid excessive shrinkage.

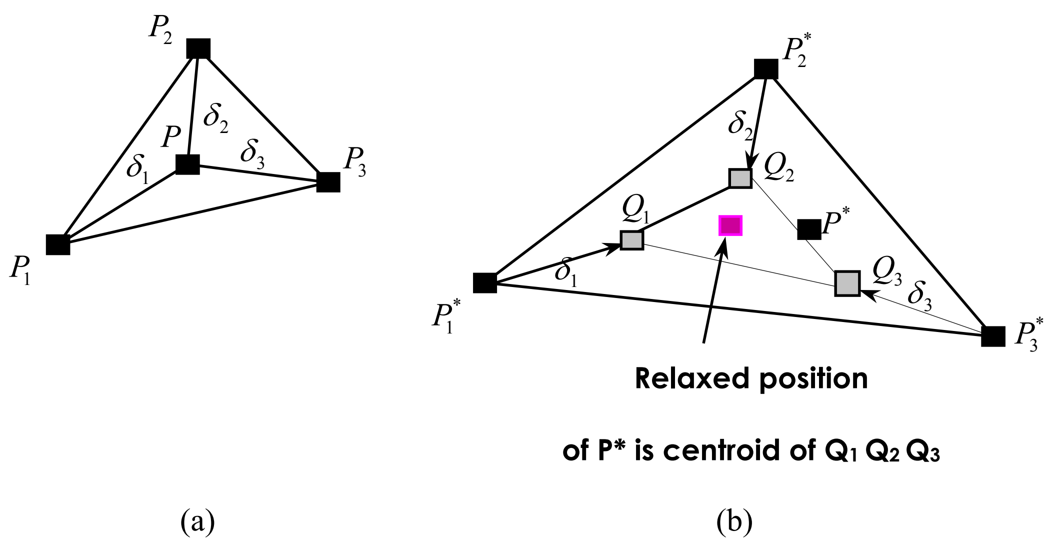


Figure 5.5: *(a) The original, undeformed state of* P, P_1, P_2, P_3*; (b) After deformation, the new positions are* P^*, P_1^*, P_2^*, P_3^* *and after relaxation,* P^* *moves to the position shown.*

Each element position is adjusted toward its optimal position in an iterative process. As a result, the global system energy is progressively reduced and the object moves to an equilibrium state.

The pseudocode for the process can be expressed as:

for(*all moved element***)**

{

define the optimal position relative to its neighbours;

move the element towards that position by step size.

}

This minimization of elastic energy gives a physical basis to the ChainMail algorithm, to complement the more intuitive, but fast, deformation part.

5.2.6 Generalised ChainMail in Other Dimensions

The algorithm for the case of a 2D planar set of linked elements has been described. This section describes how the idea translates into spaces of different dimension.

For a 3D volume, where the elements are linked in a tetrahedral mesh, one needs to make the following changes to the above algorithm. The data structure (section 5.2.1) changes to reflect the 3D dimension of object and space. The rules for updating the position of an element (section 5.2.2) change as follows. The valid region becomes a cuboid, *C*, given by:

$$C = \{(x, y, z) : x_{\min} \le x \le x_{\max};\ y_{\min} \le y \le y_{\max};\ z_{\min} \le z \le z_{\max}\}$$

where (supposing $x_2 \ge x_1,\ y_2 \ge y_1,\ z_2 \ge z_1$):

$$\begin{aligned}
x_{\min} &= x_1^* + (\alpha_{\min}\Delta x - \beta(\Delta y + \Delta z)) \\
x_{\max} &= x_1^* + (\alpha_{\max}\Delta x + \beta(\Delta y + \Delta z)) \\
y_{\min} &= y_1^* + (\alpha_{\min}\Delta y - \beta(\Delta z + \Delta x)) \\
y_{\max} &= y_1^* + (\alpha_{\max}\Delta y + \beta(\Delta z + \Delta x)) \\
z_{\min} &= z_1^* + (\alpha_{\min}\Delta z - \beta(\Delta x + \Delta y)) \\
z_{\max} &= z_1^* + (\alpha_{\max}\Delta z + \beta(\Delta x + \Delta y))
\end{aligned}$$

with similar changes to the 2D case if $x_2 \leq x_1$, $y_2 \leq y_1$, or $z_2 \leq z_1$.

The cascade of element moves (section 5.2.3) behaves exactly as in the 2D case, and the termination proof carries over straightforwardly.

Similarly the relaxation described above for the 2D case extends quite straightforwardly to higher dimensions.

The key dimension for the algorithm is the dimension of the space in which the object lies, not the dimension of the object itself. Therefore the algorithm for a surface mesh within a 3D space is exactly the same as for a volumetric mesh. This unification of surface and volume deformation is an important attribute of generalised ChainMail. Indeed the algorithm for a 1D curve object in a 3D space is similar also.

Finally one notes that the case of a 1D object in a 1D space is particularly simple as there is no shearing effect.

5.3 Results

This section reports on the practical implementation of the new algorithm. It has been implemented as a combination of Java for the simulation engine, and VRML for the display engine, with the EAI providing the communication link between the two.

The first example, see Figure 5.6, shows part of a skull modelled as a surface, with nodes connected in an irregular mesh. The surface has been extracted from CT scan data of the skull. The mesh consists mainly of triangles, but note that for the simulation we deal not in terms of triangular pieces, but rather in links between the elements of the mesh. Thus it is no problem accommodating pieces of different shape in the mesh and in this case there are some quadrilateral elements. The mesh has 345 elements and 992 links. We are able to obtain real-time interactive performance on an AMD-K7 processor, with TNT2 M64 graphics card and 128 Mbytes memory, running the CosmoPlayer viewer as plug-in for Internet Explorer 6, when the skin over the surface of the skull is deformed by pulling on the element shown (marked with small sphere).

Figure 5.6 shows a sequence of three images showing deformation progressively occurring.

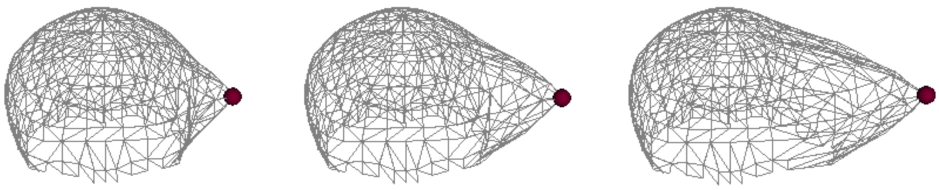


Figure 5.6: Surface Deformation using Generalised ChainMail

The second example, see Figure 5.7, shows a further surface deformation: a brain surface with a complex structure containing both concave and convex features. The model has 1043 elements, 2079 triangles and 3119 links. Frame rates greater than 45 frames per second were achieved on the system configuration described above. Figure 5.7 again shows the brain surface being stretched:

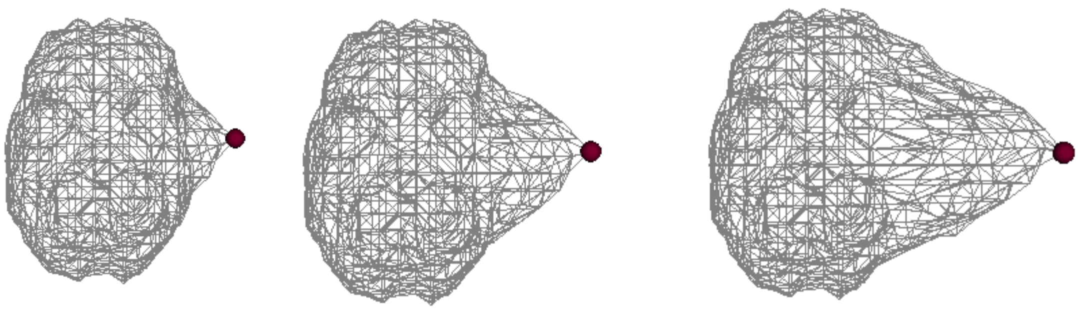


Figure 5.7: Brain Surface Deformation

The above sequences as wire frame images have been intentionally shown, in order to show the stretching of the nodes. In practice, it would of course be normal to render as shaded surfaces, and Figure 5.8 shows a rendering of the brain surface of Figure 5.7, during the deformation process.

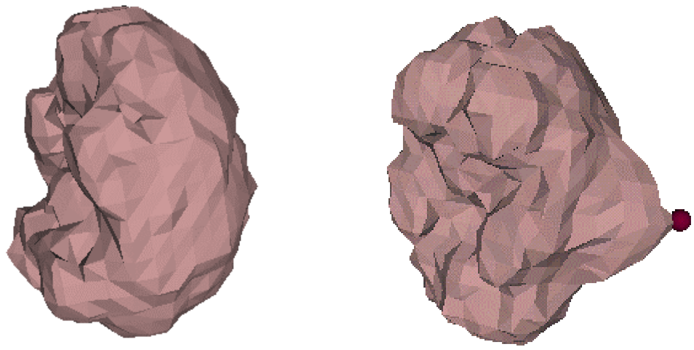


Figure 5.8: Brain Surface in Shaded Rendering Mode

Figure 5.9 shows the 3D volumetric mesh version of the algorithm. The mesh has 343 elements and 1845 links. The maximum number of neighbours for elements in this model is 14. Real time interaction is again achieved.

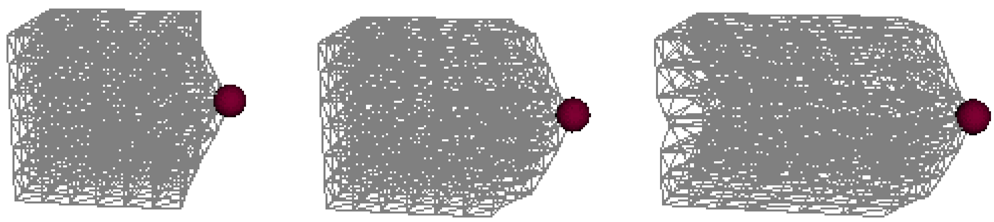


Figure 5.9: 3D Volumetric Mesh under Deformation

5.4 Discussion

An extension of the original 3D ChainMail algorithm to arbitrary meshes, of any dimension has been described. The algorithm is fast enough to be used for real-time interactive deformation of soft tissue, in a Web-based environment. It thus becomes feasible to use it as part of web-based training for surgeons: accessible anywhere, at any time, and by as many students as there are PCs available. Important aspects are that changes of topology, such as cutting, are easily achieved, and that surfaces and volumes are treated uniformly within the same overall scheme. It has applications beyond surgical training, to any deformable modelling in a Web-based environment.

The evaluation of the performance of the Generalised ChainMail algorithm together with the SurfaceChainMail algorithm will be presented in the next chapter.

Chapter 6 Evaluation

Previous chapters described the training simulator for Percutaneous Rhizotomy procedure (PRP), the simple cutting simulator and the deformable modelling techniques, the SurfaceChainMail and the Generalized ChainMail, developed in this study. The primary objectives of the cutting simulator were to investigate whether the topology changes can be made with the SurfaceChainMail and whether deformable modelling techniques can be integrated into a web-based environment to simulate the simple cutting procedure. These were achieved and shown by the outcomes of the system. As the development of the cutting simulator is still preliminary, further study on it would be needed prior to its evaluation. In this chapter, two evaluation processes are carried out: the evaluation of the PRP training simulator and the evaluation of the extended ChainMail algorithms (the SurfaceChainMail and the Generalized ChainMail). The PRP training simulator is evaluated by the qualitative and quantitative measurements of the quality of the simulator. The SurfaceChainMail and the Generalized ChainMail are assessed by comparing their performances with the original ChainMail developed by Gibson.

6.1 Evaluation of the PRP Training System

The evaluation process can be carried out in different stages of the system development. Different methods can be used at the various stages. In order to assess the success of system development, one needs to identify the criteria which are important for gauging the success. In this section, the criteria are identified, followed by specifying the method of study, and results are shown next. At the end, suggestions for further improvement of the system are made.

6.1.1 Evaluation Criteria

The evaluation of the quality of the system refers to the ISO 9126 standard (ISO, 2002), an international standard for software product evaluation. This standard defines six product quality characteristics, which are applicable to every kind of software. They are:

- Functionality – whether the software can satisfy the required functions.
- Usability – whether the software is easy to learn and use.
- Reliability – whether the software is reliable.
- Efficiency – whether the software is efficient.
- Maintainability – whether the software is easy to modify.
- Portability – whether the software is easy to use in a different environment.

These characteristics are common attributes for all software. Each individual piece of software will have particular characteristics that are more important to evaluate than others. In the virtual reality field, most work focuses on usability to quantify the effectiveness of the VR system. Usability is defined in (Bowman et al., 2002) as “ease of use” and “usefulness”, including quantifiable characteristics such as learnability, speed and accuracy of user task performance, and user error rates together with subjective user satisfaction and user comfort. Three kinds of usability evaluation can be profitably applied to virtual environments: expert heuristic evaluation, where only the usability experts are involved in the evaluation; formative evaluation, where the potential users are involved to produce qualitative and quantitative results; and summative evaluation, where the users of the system are compared with the users of another system for performing the same user tasks (Hix et al., 1999). In order to identify usability problems in both formative evaluation and summative evaluation, quantitative and qualitative data are collected from the representative users with task-based scenarios.

There are many factors that affect the user performance for a VR-based system. For example, experimental evidence shows that the distance from the user to the object being selected and the size of the object affect the user’s performance in selection tasks significantly (Bowman et al, 2001). Frame time variation (Watson et al., 1997) is another factor that affects the user performance. However, it is often difficult to know which factors have a potential impact on the results when performing formal experiments to quantify and compare the usability of various virtual environments, interface elements, and so on (Bowman et al., 2002). One of the solutions to this problem suggested by Bowman et al. (2002) is to hold as many of these other factors as possible constant, and evaluate only within a restricted set of circumstances. Therefore, a number of researchers focused on some of the potentially important effects for certain tasks. For example, Watson et al. (1997) set up an experiment to examine the effects of frame time variations, in both deviation around the mean frame time, and the period of fluctuation, on a task performance in a virtual environment. The experiment was performed on participants with different backgrounds in using virtual reality and head-mounted displays. The authors looked at the manipulation tasks such as grabbing and placement, and measured both accuracy and time for performing these tasks. The results pinpointed the range of acceptability for frame time fluctuations for VR-based designers.

Some researchers attempted to incorporate an assessment tool as part of their VR-based surgical training systems to obtain the objective measurements of task performances. They used large resources, in terms of time and money, to identify the usability problems. Common experimental methods used in the investigation utilised two groups of participants (e.g. Smith et al., 1999). One was a novice group and the other was an expert group. When the experienced surgeons performed surgical tasks significantly better than the novices on a surgical training system, it was claimed that the simulator might be useful in quantifying surgical skills. Recently, Torkington et al. (2001) validated the usefulness of virtual reality surgical simulators, by studying the transfer of skills by assessing their performance on real laparoscopic tasks. The test was carried out between three different groups of users for minimal invasive surgery: one with no training, one with the MIST-VR trainer and the other one with conventional training. Each group then underwent a post-test using the Imperial College Surgical Assessment Device. The results showed that the training of novices using MIST-VR group had quantifiable changes in skills that were transferable to a simple real task and were similar to the results achieved with conventional training.

6.1.2 Evaluation Framework

The evaluation involved in this study was an application-specific context instead of a generic context. The expert heuristic evaluations were carried out throughout the development of the PRP system. The recommendations from the experts were collected and early development of the system was modified according to their suggestions. For example, direct manipulation of the virtual needle early in the development of the PRP system was found too difficult to handle, and so it was modified by using widgets and placed together with other controllers in one VRML world.

Further evaluation was carried out on the initial prototype system and recommendations for improvement of the system performance were obtained from the user group and experts for the future development. The evaluation was partitioned into two parts: VR-related and medical-related.

For the VR-related evaluation, the representative users were defined as any potential users of a VR-based system. In order to assess how well the interface of the PRP simulator supports the functions for particular tasks specified in the system, the main features (e.g. the reference planes and the local viewer from the needle) were examined by evaluating user task performances with the tasks being: needle locating, rotating, and inserting to produce qualitative and quantitative results. The performance metrics included the speed of completion of the tasks, the accuracy of the performance for the tasks, and the ease of using and learning of the simulator.

The accuracy measurements are obtained from the assessment tool – the error rates recorded when users performed the procedure with the simulator. The comparison of the results that were generated by performing the procedure with and without the features being assessed, are given at the end of this section. As only two performance metrics were tested, these quantitative results cannot fully cover all relevant important aspects of a specific task. Therefore, a post-session questionnaire was given to acquire some of the qualitative information that was associated with the usability issues, as a complement to the important aspects of a task. A combination of the quantitative measurement and qualitative measurement through the participants - formative evaluation, is the major focus in this evaluation.

On the other hand, the representative users in medical-related evaluation were defined as a neurosurgeon. The goal of this part of the evaluation is to identify the usability problems and collect the suggestions in improvement for further development from the potential users of the simulator. Because of the lack of resources, such as time and money, in this study, this part of the evaluation was simply carried out based on the expert heuristic: interviews with an experienced neurosurgeon who is also an expert in virtual reality field.

6.1.3 Effects on Performance

In the PRP procedure, the key tasks include the location, rotation and insertion of the needle. We have only focused on these tasks and ignored the other common tasks such as navigation and selection of objects. This is mainly because most functions provided for these other tasks are given by the standard interface of VRML browsers.

In order to perform these tasks easily, in the design of the PRP system, additional features were added to help users to determine the location and insertion of the needle. These features include the local viewer, the reference planes, the transparent skin and the planes through the landmarks. We need to examine whether each of these features has an effect in improving performance results. In addition, collision detection techniques were developed so that users can "sense" whether or not the needle has hit the targets when performing the tasks. This was also included in the evaluation to consider whether it is a useful characteristic for task performance in this application.

6.1.4 Method

In order to get meaningful results, a two-trial scheme was used: informal-tests and formal-tests. The informal-tests were used to determine which measurements defined in the experiments were important for evaluating the performance in the experiments, and which methods can be most effectively used to examine each of the usability problems. The formal-tests have experiments selected from the informal-tests. Thus, the informal-tests took more experiments, and therefore took longer to complete than the other.

In the informal-tests, five participants were selected from outside the department. Like the formal-tests, they performed the tasks with different features such as local viewer and reference planes. As timing is regarded as an important factor in assessing the performance in the informal-tests, their time was recorded during the experiments. Two tests for the location of the needle were performed: locating the landmarks with and without reference planes respectively. Each of the tests was repeated three times.

Five tests for the rotation and insertion of the needle were performed: rotating and inserting the needle with and without the planes through landmarks, with and without the local viewer, and without collision signals. Again, each of the tests was repeated three times.

The results showed that the time taken by individual participants varied depending on the individual characteristics of the participants. It was therefore decided not to evaluate the performance of the time in the experiment. It was observed that towards the beginning of the experiments, the participants had less experience and were not very familiar with the simulator. However, by the end, they had more practice and became quite competent. Therefore, the order of the tests is important. The results showed that if the test for the needle rotation and insertion tasks with the planes through landmarks was arranged at the beginning of the experiment and the test for performing the tasks without any clues was arranged at the end of the tests, these two results were closer than mixing the tests together. To achieve a fairer test, it was decided to go through each of the tests consecutively and repeat this process twice more for the rotation and insertion tasks in the formal-tests. Further, these tests often took a lot of time, and so the participants grew tired and bored towards the end. This meant that they might hastily finish off the tests, and the results might not be accurate. Therefore, the tests for the planes through the landmarks were reduced, while keeping other important features unchanged, to minimise the time taken during the experiments. However, the usefulness of this feature was included in the post-session questionnaire completed by the participants.

In the formal-tests, the participants were 6 volunteers, (1 female and 5 males), who were selected from within our school and had different experiences in using VR-based simulation systems and VRML browsers. Their average age was 30. Before the experiments started, the participants were given pre-session background questionnaires, which were designed to obtain general information from the participants. This was then used to analyse the other factors that may have an effect on performance. They were then requested to go through the web pages explaining the tasks and how to use the simulator. A brief introduction was given to the participants to help them understand the tasks. Each participant played with the simulator a few times to get used to the environment before the experiments started.

During the experiments, the evaluator observed each individual performing the tasks, and made notes about each of the incidents occurred when they performed the tasks. Each participant was requested to perform the tasks under several conditions, and to perform three times with each condition. All participants were requested to use the same VRML browser - CosmoPlayer.

- Location of the needle

The reference planes were tested to see whether they are useful for locating the needle. During the tests, the other features were made invisible to the user. The participants were firstly asked to make landmarks on the face in the global viewer without the reference planes. Then the planes were made visible to the user and the tests were repeated three times. The results were recorded by the assessment tool automatically.

- Rotation and insertion of the needle

Before starting the tests for the rotation and insertion tasks, the initial position of the needle has to be determined. In order to get meaningful results, it must be reset exactly the same in each test. This was approximated by referring to the reference planes by the evaluator. The results of the location were checked from the assessment tool. If it is within the specified range, the test is continued. Otherwise, it will be repeated until a satisfactory result is obtained.

Two features were tested for the tasks: the local viewer and the collision signals. The local viewer was tested by showing or hiding it from the participants. The collision signals were tested by making the signals invisible to the participants, but visible to the evaluator. The evaluator very carefully observed the participant performing the tasks to see whether they stopped or continued the insertion task when the needle hit the skull or target, or stopped the insertion task before the needle hit the objects. Finally, all the participants performed the tasks without any of these visible clues. The mean errors were obtained by averaging their results.

At the end, the end-of-session questionnaire was given to the participants to provide qualitative evaluation by using a rating scale for each component, and the suggestions of improvement for future development. The rating scale is a scale ranging from 1 to 10, with 10 being the best and 1 being the worst.

6.1.5 Results and Discussion

Table 6.1 shows the overall results from all the participants. For the location task, three landmarks were measured. The ideal position for each of the landmarks should be:

Mark 1: 2.5cm from the angle of the mouth within the axial plane;

Mark 2: below the pupil within the saggital plane;

Mark 3: 3cm from the ear within the coronal plane;

Suppose *Dist1*, *Dist2*, and *Dist3*, are the distances from the marking position to the angle of mouth, pupil and ear respectively. *E* is the error for the task; and

$D1 = |Dist1 - 2.5|$; $D2 = Dist2$; $D3 = |Dist3 - 3|$.

Thus, the error for this task for each subject was calculated as:

$$E = ((D1 + d1) + (D2 + d2) + (D3 + d3)) / 6$$

where *d1, d2,* and *d3,* are the distances from the marking position to the axial plane through the angle of the mouth, saggital plane through the pupil and axial plane through the ear respectively. The error for the rotation and insertion tasks was calculated as $E = Dist - R$; where *Dist* is the distance from the centre of the foramen ovale to the tip of the needle; *R* is the radius of the foramen ovale. All the measurements were scaled to real distances. The mean errors were the average over all the subjects.

Table 6.1: Performance test results

Results	**Location**		**Rotation and Insertion**	
	reference planes	without clues	local viewer	without clues
Mean error (cm)	0.32	0.47	3.2	4.4

In the needle location task, the results with the reference planes were slightly better than the results without those planes. However, the participants gave very high scores (average 9.5) for the planes in the post-session questionnaire. A possible reason is that, when performing the task, the planes blocked the participant's view, and thus made locating the needle quite difficult, unless the angle of the view was carefully shown. As the planes can be made invisible easily, it might be a good idea to switch on the planes to get a general idea of where the marks should be placed and then switch them off to continue the marking task. Alternatively, it may be more useful if the planes are used for an off-line analysis for the user to get better results. For example, the user can compare the result that they made with the planes after performing the task, and they could get ideas where each of the landmarks should be.

The local viewer was shown, from both results – quantitative results and qualitative scores (average 9.7), very useful in improving the performance for the rotation and insertion tasks. The ratings for the tasks with the planes through landmarks were very positive (average 8.8 in the post-session questionnaire). It was found that it would be more useful if it was used as an off-line analysis, just as the reference planes.

The collision signals were tested by setting the signals invisible to the participants but visible to the evaluator, by covering it with an opaque sheet of card, and angling it so that the participant could not see the collision signals, but the evaluator still had a line of sight to it. It was found that in most tests, the participants still continued the insertion movement when the needle hit the skull or the foramen ovale. In a few tests, they stopped the movement before the needle hit the skull. One feature that was found by a participant was that, when the needle hit the skull, the needle bounced back and was thought to be an additional useful signal to the collision. The qualitative scores for the signals were also very positive (average 9.3).

The participants found that the web page describing how to use the simulator, the interface, and 3D Graphics were very easy to understand. They found the controllers quite easy to use but not the widget for the rotation in the vertical direction. It was observed that one participant manipulated this widget vertically, which was not correct. The participant eventually got used to it after a few tests. Suggestions for the improvement of the system include using a beeping collision signal instead of lights, and using handles to control the rotation instead of wheels.

An interview with a consultant neurosurgeon was conducted after having watched him performing the surgical tasks three times. He believes that the simulator would be very useful for a range of people, such as medical students and neurosurgeons who have not had experience with the procedure. It would also be very useful for experienced neurosurgeons to teach the trainees, and to assess their performance with the assessment tool. They should be able to use the simulator once they have practiced with it a few times. The Web-based environment provides advantages of independence of both time and place, which will be very useful for their self-learning. In addition to the benefit of performance assessment for the experienced surgeons, it will be also very useful for the trainees to improve their performance by showing their results displayed on the screen. He suggested that the widget for the rotation controls in vertical direction should turn a 90 degrees for ease of use. He was very happy with the additional local viewer which helps a user to perform the tasks. However, he also suggested that the interface is too complicated and should be simplified to make the tool more readily accepted by experienced neurosurgeons who typically have less patience in learning new things!

6.2 Evaluation of the Extended ChainMail Algorithms

3D ChainMail developed by Gibson is a promising approach for real-time deformation. Its fundamental difference from other approaches is that real time performance can be achieved with large datasets and it can support large deformation and real time topology changes. These advantageous features have been demonstrated in (Gibson, 1997; Gibson et al., 1999). In this research, the extensions to the Gibson's ChainMail, SurfaceChainMail and Generalized ChainMail, were proposed. However, it is still uncertain how good the proposed approaches are, comparing with Gibson's ChainMail. In other words, we still need to know whether the new ChainMail approaches preserve the fast speed of deformation, or sacrifice this feature in order to support the greater generality of data structure. To address this problem, a performance evaluation is carried out and described in detail in this section.

6.2.1 Methods

To examine the performance of the new algorithms, a comparison was performed between the Gibson's ChainMail, SurfaceChainMail and the Generalized ChainMail. The speed of the algorithm was measured by the frame rate of a VRML browser when the displayed model is deformed. The measurements were carried out on a number of 2D systems. The test data used for each algorithm were 2D objects with the following dimensions: 11x11, 25x25, 51x51 and 75x75 (elements). In order to make direct comparison of the performance between the algorithms, the meshes of the objects are all rectangular grids, although the new algorithms can also be applicable to much more general grids. In each experiment, objects with different softness, rigid, medium and soft, were chosen. The medium and soft objects were defined as in (Gibson, 1999): The medium object allows link lengths to be compressed to 80 percent of the optimal length and stretched by a factor of 1.2 from the optimal length; The soft object allows link lengths to be compressed to 50 percent of the optimal length and stretched by a factor of 2.5 from the optimal length. The amount of displacement of the controlled element varied during the test. The test was performed on a PC with AMD-K7 processor, with TNT2 M64 graphics card and 128 Mbytes memory and the measurements were made on Microsoft Internet Explorer 6.0 with CosmoPlayer 2.1.1.

6.2.2 Comparisons of Performance

The intention for the performance test was to compare the relative speed of the three algorithms. Our interest is focused on web-based environments and the evaluation is made in this context: Java is used for the modelling calculations and VRML for the display (using the CosmoPlayer viewer). It would be expected that the relative performances of the three algorithms are similar in non-web environments, say a direct implementation in terms of C with OpenGL, where of course one could anticipate even better absolute performance.

Tables 6.2 to 6.4, together with Figures 6.1 to 6.3, display the testing results. The range of frame rates was measured when the element in the centre of the object was picked and displacements were changed from small movement (only the neighbouring elements are involved in the movement) to large movement (all the elements in the objects are involved in the movement). Frame rates are provided by both CosmoPlayer and Cortona browsers through an API call (or keyboard shortcut) – in these tests, CosmoPlayer was used. The frame rates were measured for a range of displacements of a target node: the highest and lowest frame rates (high for small displacement and low for large displacement where many nodes are affected) were recorded. The worst-case scenario for each object is when the object is rigid, and all the elements of the object are considered and moved for each displacement.

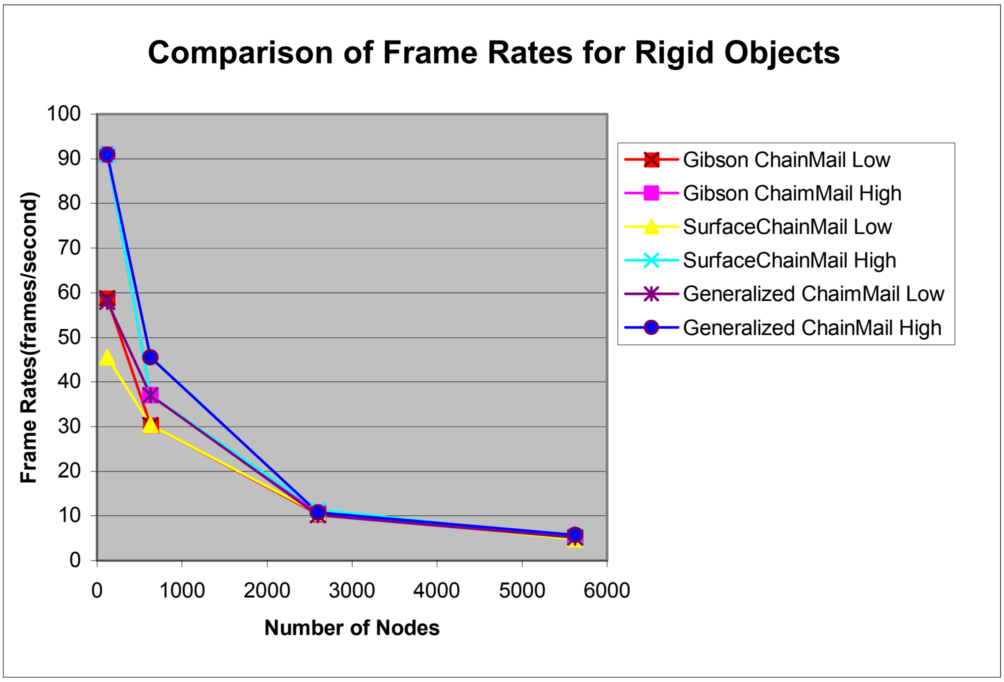


Figure 6.1: Comparison of Frame Rates for Rigid Objects

Table 6.2: Comparison of Frame Rates for Rigid Objects

Dimension (nodes)	**Frame Rates** (frames/second)					
	Gibson ChainMail		SurfaceChainMail		Generalized ChainMail	
	highest	lowest	highest	lowest	highest	lowest
121	90.9	58.82	90.9	45.5	90.9	58.82
625	37	30.3	37	30.3	45.5	37
2601	10.75	10.36	11.4	10.6	10.8	10.1
5625	5.2	5.18	5.2	4.8	5.7	5.1

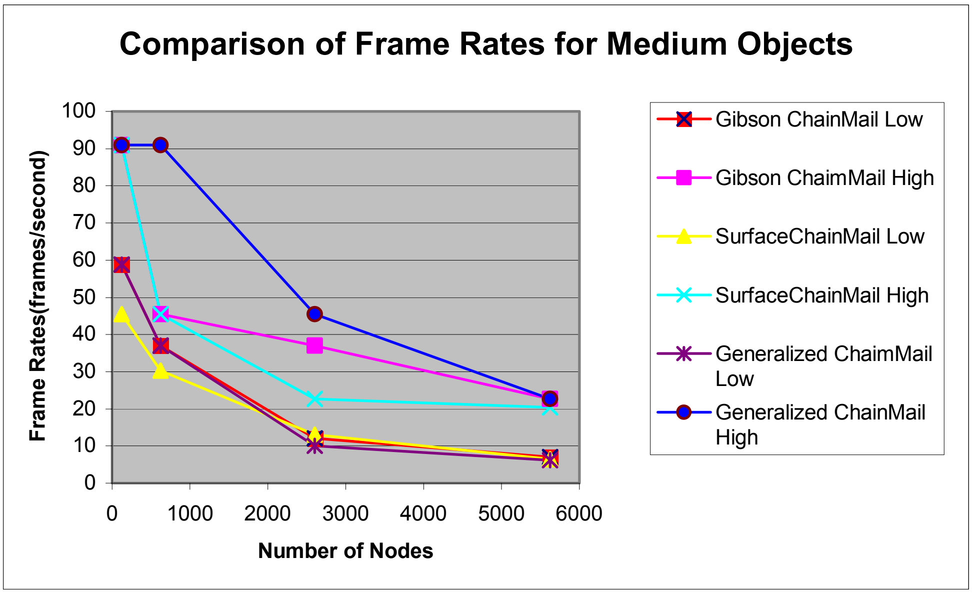


Figure 6.2: Comparison of Frame Rates for Medium Objects

Table 6.3: Comparison of Frame Rates for Medium Objects

Dimension (nodes)	**Frame Rates** (frames/second)					
	Gibson ChainMail		SurfaceChainMail		Generalized ChainMail	
	highest	lowest	highest	lowest	highest	lowest
121	90.9	58.82	90.9	45.5	90.9	58.82
625	45.45	37.03	45.45	30.3	90.9	37
2601	37.03	12.04	22.7	13	45.5	10.1
5625	22.73	6.99	20.4	6.5	22.7	6.1

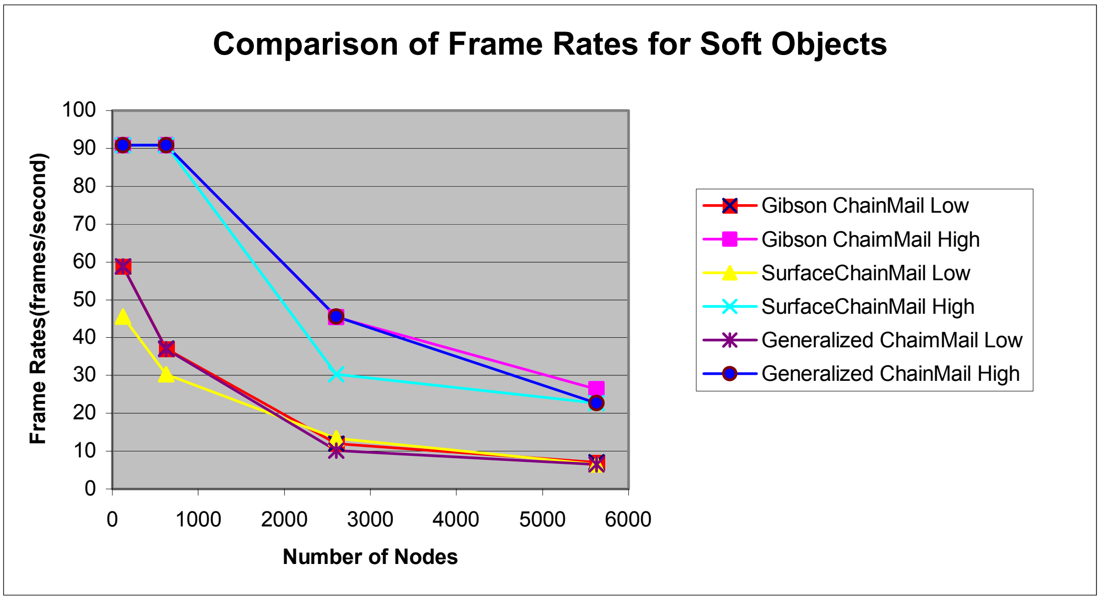


Figure 6.3: Comparison of Frame Rates for Soft Objects

Table 6.4: Comparison of Frame Rates for Soft Objects

Dimension (nodes)	**Frame Rates** (frames/second)					
	Gibson ChainMail		SurfaceChainMail		Generalized ChainMail	
	highest	lowest	highest	lowest	highest	lowest
121	90.9	58.82	90.9	45.5	90.9	58.82
625	90.9	37.03	90.9	30.3	90.9	37
2601	45.45	12.04	30.3	13.4	45.5	10.1
5625	26.31	6.99	22.7	6.5	22.7	6.5

The experiments show that the new algorithms are equally as fast as Gibson's ChainMail. The SurfaceChainMail model for the simple case is close to the Gibson's ChainMail. However, the shape of the object in SurfaceChainMail affects the performance in speed. The data tested in this experiments were the simplest cases – the 2D surface deformed in 3D space. When the parameterisation process is complicated, the performance can be even worse. For example, if the object is a cylinder with dimension of 11x11, the frame rates are between 5 and 63 frames per second, which are much lower than the other two algorithms. The frame rates decrease when the sizes of the objects increase with all of the algorithms. When the displacement of the selected element is small, the softer the object is and the higher the frame rates are, as a smaller number of elements are involved in the movement. The results of the worst-case scenario with the Generalised ChainMail, for the case of rigid objects, are shown in Figure 6.4. Each experiment was performed by moving an element located at the centre of the object at a constant rate, twenty times around a square path with dimension of 40x40, which is similar to the experiments in (Gibson, 1999). The results show that the time taken to deform an object, at worst, scales almost linearly with the number of elements in the system. This reflects the important property of the Generalised ChainMail algorithm whereby one only needs to adjust each element once in an iteration. This is similar to the results produced by Gibson's algorithm which has a similar property.

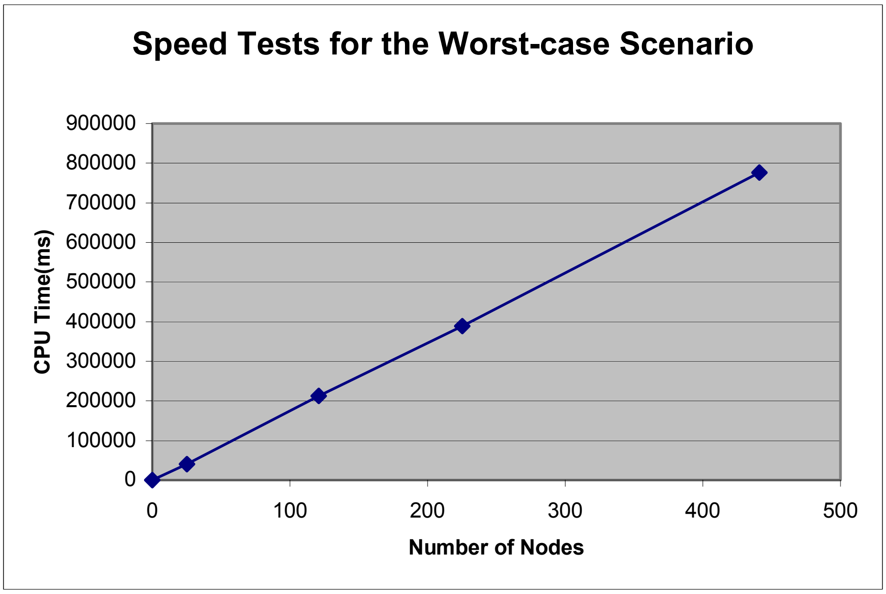


Figure 6.4: Speed Tests for the Worst-case Scenario

Table 6.5 is the performance, in terms of speed, of a list of recent methods that have attempted to achieve real time deformation and cutting. The middle three were selected from (Monserrat & Hernández et al., 1999) and the last one was obtained from (Neinhuys et al., 2001). These results were selected from the published papers, and so the results apply to the different platforms shown. Although the Generalised ChainMail algorithm was tested in a web-based environment with consequent performance penalty induced by Web technologies, it still achieves very pleasing results.

Table 6.5: Comparison of real time deformation techniques

Models	Number of Nodes in Real Time	Platforms
Generalised ChainMail	5600	AMD-K7 processor, with TNT2 M64 graphics card and 128 Mbytes memory, running the CosmoPlayer viewer as plug-in for Internet Explorer 6
Monserrat (BEM)	150	Silicon Graphhics workstation with an R-4400 processor and 64Mbytes of RAM
Delingette (mass-spring)	600	Silicon Graphhics workstation with an R-4400 processor and 64Mbytes of RAM
Terzopoulos (the Spline physic)	64	Silicon Graphhics workstation with an R-4400 processor and 64Mbytes of RAM
Neinhuys (Fast FEM)	2000	500 Mhz Xeon CPU

6.3 Discussion

3D ChainMail is an attractive solution to real time deformation for large datasets. It is however restricted by a number of assumptions: the grid must be rectilinear; and the algorithm applies only to planar (when treated as a 2D algorithm) or volumetric objects. In this research, extensions are proposed with the aim of gaining wider applicability but still retaining real-time performance. The experiments carried out showed that the performance, in terms of speed, of the SurfaceChainMail is close to Gibson's ChainMail when the parameterisation process from a 2D plane to the modelled object is simple. However the performance deteriorates as the parameterisation process becomes more complex. This approach can only be used for surface objects represented as quadrilateral meshes. This has been further extended to the Generalised ChainMail for both surface and volumetric objects represented as arbitrary meshes. The timing experiments show that it has similar results to Gibson's ChainMail on a variety of grid sizes and material properties. However the ability to handle arbitrary surface and volumetric meshes widens the use of the approach to a greater class of applications – without any decrease in performance.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

Virtual Reality (VR) has great potential for surgical training and performance assessment although it typically requires dedicated and expensive equipment. Web-based VR offers an innovative approach for group or distance learning and training. The requirements of the equipment used by the Web-based VR systems are modest – an Internet-connected computer or a small workstation with a web browser - and the simulation can be accessed worldwide.

In the collaboration with the neurosurgeons at Leeds General Infirmary, a Web-based VR system as a tool of training neurosurgeons in Percutaneous Rhizotomy (a treatment for the intractable facial pain which occurs in trigeminal neuralgia) has been developed. This involves the insertion of a needle so as to puncture the foramen ovale, and lesion the nerve. The measurement of performance can be recorded by the embedded assessment tool.

The surgical simulation uses VRML to provide a 3D visualization environment, but the work immediately exposes a key limitation of VRML that it does not support collision detection between objects - only between viewpoint and object. Thus collision between needle and skull (or foramen ovale) cannot be detected and fed back to the trainee. A novel solution has been proposed and investigated in this study. The training simulator has linked views: a normal view, plus a view as seen from the tip of the needle. Collision detection is captured in the needle view, and fed back to the normal viewer. A promising consequence of this approach has been the chance to aid the trainee with this additional view from the needle tip, which helps locate the foramen ovale. The technology to achieve this is Java software communicating with the VRML worlds through the External Authoring Interface (EAI).

The usability studies of the training system using a combination of heuristic and formative evaluation have been presented and shown very positive results. This simulator has been endorsed as a useful training and assessment tool for a range of people, such as medical students, neurosurgeons who have not had experience with the procedure, and for experienced neurosurgeons to teach the trainees and to assess their performance with the assessment tool. A Web-based environment is considered very useful for their self-learning. The training simulator is available on the Web, with accompanying tutorial on its use.

The modelling of soft objects is another objective in this study and crucial in providing realistic simulation of many surgical procedures. High accuracy is achievable using the Finite Element Method (FEM), but significant computational power is required. We are interested in providing Web-based surgical training simulation where such computational power is not available, but in return lower accuracy is often sufficient. A useful alternative to FEM is the 3D ChainMail algorithm that models elements linked in a regular, rectangular mesh, mimicking the behaviour of chain-mail armour. An important aspect is the ability to make topology changes for example by cutting – an aspect that FEM finds difficult. The 3D ChainMail algorithm has been further developed and modified in this research. Two extended algorithms, the SurfaceChainMail and the Generalized ChainMail, have been developed. The SurfaceChainMail extends the algorithm to non-uniform rectilinear meshes and to surface-based representation of objects. It has been implemented as a Web-based application to build a general simulator for the cutting and separation of layers of soft tissue. This simulator has demonstrated the feasibility of achieving real time topology changes on the Web with the SurfaceChainMail.

A key restriction of the SurfaceChainMail has been the need to use meshes which are rectilinear. Therefore, the Generalized ChainMail has been developed to handle arbitrary meshes in 2D and 3D. This extends the range of applications that may be addressed by the ChainMail approach, to include surfaces and volumes defined on triangular and tetrahedral meshes. The action of processing each element of objects no more than once is justified and the details have been presented. The modified algorithm has been successfully deployed in a Web-based environment, using VRML and Java linked through the External Authoring Interface. Real-time interaction with the model has been achieved.

A number of experiments have been undertaken and shown that the speed of the SurfaceChainMail model for the simple case is close to Gibson's ChainMail and the speed of the Generalised ChainMail is equally as fast as Gibson's ChainMail. An important aspect with the Generalised ChainMail is that surfaces and volumes are treated uniformly within the same overall scheme: a surface represented as a triangular mesh in 3D space is deformed using the same algorithm as a volume represented as a 3D tetrahedral mesh.

Although the approaches investigated in this research have been applied in surgical training applications, they can be used in Web-based applications beyond surgical training.

7.2 Future Work

In this thesis, the Web-based training system with the assessment tool has been presented. According to the suggestions from users, the interface can be improved to make it easier to use. As a training tool, the evaluation of skill transfer from the training tool to reality, using more formal methods, is necessary for further investigation. A possible way to do that is to test on two groups of surgeons: the first group with the conventional training method and the second group with the additional method, the surgical training tool. If the skills are transferred, the second group would perform better than the first group. Another evaluation is necessary to test whether the assessment tool can be used to distinguish different levels of performance between surgeons. Again, this can be tested on two groups of surgeons: trainees and experienced surgeons. If it is useful, a corresponding result must be the experienced surgeons have higher scores than the trainees when they use this tool.

For deformable modelling techniques developed in this work, it does not claim to provide the high accuracy of FEM, but this is less important than real-time interaction in our application. At present, the scheme is restricted to homogeneous materials – it would be nice to extend it to inhomogeneous material (as was done for the original ChainMail in Schill et al., 1998). Another limitation is the lack of volume preservation which can be an important property in some applications.

Appendix A: Pre-session Questionnaire

Please provide us the general information below. Any comments are welcome.

1. Are you:

   a. male b. female

2. How old are you? ___________________________

3. Your occupation is ____________________________

4. How often do you use a computer?

   a. never b. some times c. very often

5. Have you ever used any VR-based simulation system?

   a. yes b. no

6. Have you used VRML browsers?

   a. yes b. no

7. If yes, which browser(s)?

   a. CosmoPlayer b. Cortona c. Other ___________

Appendix B: Post-session Questionnaire

1 = poor

10 = very good

Understandability: (Are they easy to understand?)

Web pages:	1	2	3	4	5	6	7	8	9	10
Graphic user interface:	1	2	3	4	5	6	7	8	9	10
3D graphics:	1	2	3	4	5	6	7	8	9	10

Learnability: (Are they easy to learn?)

Graphic user interface:	1	2	3	4	5	6	7	8	9	10
3D graphics:	1	2	3	4	5	6	7	8	9	10

Operability: (Are they easy to use?)

Graphic user interface:	1	2	3	4	5	6	7	8	9	10
2D Mouse:	1	2	3	4	5	6	7	8	9	10
Controllers:										
Transparent Skin:	1	2	3	4	5	6	7	8	9	10
Validated Plates:	1	2	3	4	5	6	7	8	9	10
Plates through Landmarks:	1	2	3	4	5	6	7	8	9	10

Insertion: 1 2 3 4 5 6 7 8 9 10

Rotations: 1 2 3 4 5 6 7 8 9 10

Usefulness: (Are they useful for performing marking task?)

Validated plates: 1 2 3 4 5 6 7 8 9 10

Usefulness: (Are they useful for performing rotation and insertion tasks?)

Plates through the landmarks: 1 2 3 4 5 6 7 8 9 10

Local viewer: 1 2 3 4 5 6 7 8 9 10

Signals: 1 2 3 4 5 6 7 8 9 10

Usefulness: (Are they useful for 3D understanding?)

Transparent skin: 1 2 3 4 5 6 7 8 9 10

Local viewer: 1 2 3 4 5 6 7 8 9 10

Improvements: (Any suggestions for the improvement of the system are welcome)

1)____

____

2)____

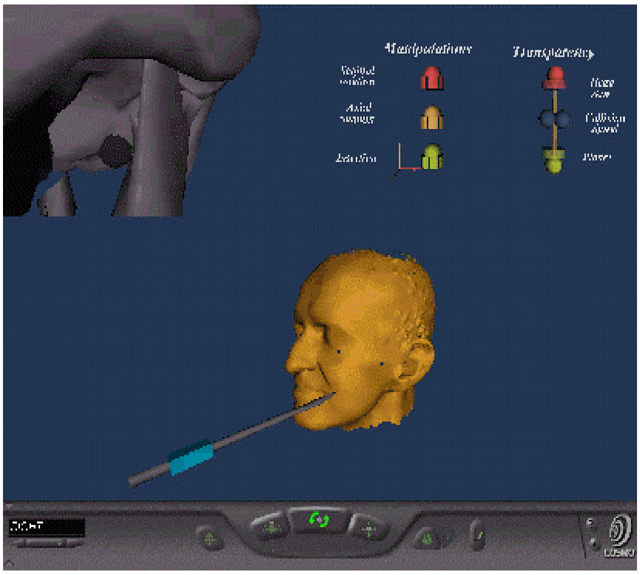
____

Appendix C: PRP Surgical Training System Web Pages

1. Introduction

Web-based Surgical Simulation:

Treatment of Trigeminal Neuralgia



Ying Li
School of Computing, University of Leeds
Supervised by:
Ken Brodlie
and
Nick Phillips,
Leeds General Infirmary
October 2002

This web-based simulator is intended to help train surgeons in the treatment of trigeminal neuralgia. It simulates the insertion of a needle in Percutaneous Rhizotomy.

It may also help a patient and their family to have a better understanding of the procedure by looking at this virtual operation.

This work is part of wider research into the use of web-based virtual environments for surgical simulation.

2. Medical background

What is Trigeminal Neuralgia?

- Trigeminal Neuralgia Association
- Jeremy S. Melker - Trigeminal Nerve Anatomy
- Facial Neuralgia Resources
- National Institute of Neurological Disorders and Stroke
- Center for Cranial Nerve Disorders and Microvascular Surgery
- University of Florida
- NMC Specializing in Surgery for Brain & Spine Disorders

What are the treatments?

- Facial Neuralgia Resources - Treatments For Facial Neuralgia
- UCSD Division of Neurosurgery
- Pain Medicine
- Adult Neurosurgery and the Beth Israel Medical Center Institute for Neurology and Neurosurgery
- The OnLine Journal of Dentistry and Oral Medicine
- Trigeminal Neuralgia Evaluation & Treatment
- South Saskatchewan Management and Research Unit

What is Percutaneous Rhizotomy?

- CCND Winnipeg
- iMigraine.net by Troost
- The Canadian Journal of Neurological Sciences
- Jho Institute for Minimally Invasive
- The University of Iowa College of Nursing - Neurosurgery Healthcare Professional Version

3. How to use the simulator

Percutaneous Rhizotomy

- *Introduction to the simulator*
- *Procedure*

Introduction to the simulator

Screen Layout | Manipulators | Collision Signal and Transparency Control

---

Screen Layout - Three Views:

The display contains three different views: Local View, Control View and Global View (see Figure 1).
Local View is the view from the tip position of the needle.
Control View provides widgets to manipulate the needle and other aspects.
Global View is the view as seen by the surgeon.

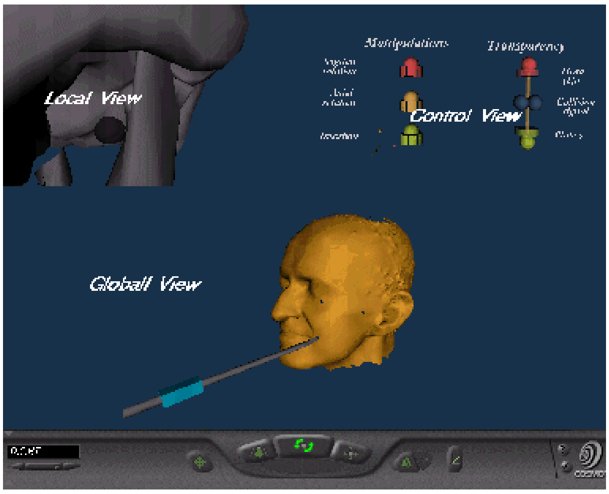
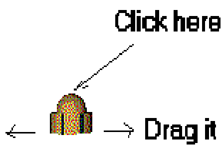


Figure 1

✸Manipulators:

In the *Control View*, these three widgets,  play different roles but are used in the same way. They are used to control the rotations and translation of the needle. To manipulate them, click on the hemisphere and release the mouse button, then drag the cylinder horizontally.



✸Collision signal and transparency control:

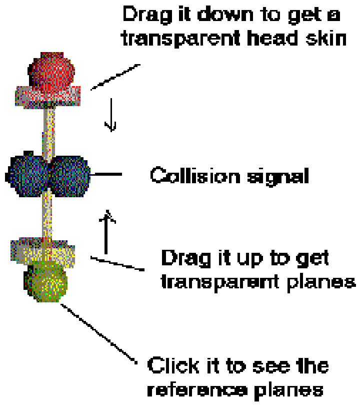


Figure 2

Apart from the three manipulators above, there are other useful widgets in the *Control View*. The transparency of the skin, and of planes through landmark points, are controlled by moving the discs up and down. The middle spheres signal that collision between the needle and skull has happened (The colour of the right sphere is changed to red) or the needle has reached the foramen ovale (The colour of the left sphere is changed to green). The green sphere toggles display of a set of ideal reference planes.

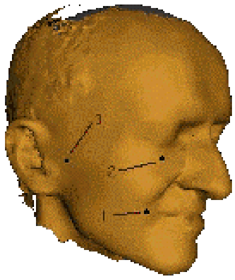
Note:
If you resize the window, or need to reload, press **Shift** + **Reload.**

Procedure

Landmarks Validation Manipulation Collision

The aim of the procedure is to puncture the foramen ovale.

Step 1. Landmarks:



Make three anatomical landmarks by clicking on the face (use the LEFT or RIGHT viewpoint):

1 The insertion point - 2.5cm from the angle of the mouth.
2 A point beneath the medial aspect of the pupil.
3 A point 3 cm anterior to the external auditory meatus.

The surgeon envisages three planes through these points: Axial through 1, Saggital through 2 and Coronal through 3. You can see the planes by dragging the green disc up (see figure 2). The intersection of the Saggital and Coronal planes guides the surgeon in inserting the needle.

Step 2. Validation:

The next step is to check the accuracy of the landmarks. Use the *Control View* to display the reference planes by clicking on the green sphere (see figure 2). If your planes are too far from the reference planes, click on the face to remove the landmarks and return to step 1. If successful, also click to remove landmarks, and proceed to step 3.

Step 3. Manipulation of Needle

Use the *Control View* to manipulate the needle. First rotate the needle in the axial plane (*axial rotation*). Second rotate the needle vertically (*saggital rotation*). Then insert the needle by manipulating the *Insertion* button. If collision happened, the needle is forced backwards. To start doing insertion again, just click on the green sphere once again. However, the needle can not be rotated after the insertion. Try to insert the needle slowly when it is closed to the skull.

The movement of the needle can be observed in the *Local View*. Alternatively the skin may be made transparent in the *Global View* by manipulating the slider in the *Control View* (see figure 2).

Step 4. Collision

If the needle hits bone, then a red collision warning light is shown on the control view. Otherwise if the foramen ovale is successfully punctured, then a green light is flashed. Click on the *Collision signal* to clear the warning light.

Where to Download & How to Navigate in Cortona?

You can download Cortona from here:

http://www.parallelgraphics.com/products/downloads

The instructions of how to navigate in Cortona can be found at:

http://www.parallelgraphics.com/developer/products/cortona/help/

4. How the simulator was developed

This simulator uses the Virtual Reality Modelling Language (VRML) to create virtual environments for surgical training and education. Three VRML browsers are embedded in the web page. They interact with each other using JAVA External Authoring Interface (EAI). Trainers and trainees can use this simulator to manipulate a virtual surgical tool in the 3D scene by using a 2D device - a mouse controller.

1. Object Modelling

VRML is the 3D modelling standard for the Internet which provides an easy to program polygonal modelling language that enables 3D visualization of remote data. In this simulation, the model of the skull and the skin are provided by Nigel John of University of Manchester. Those models combining with other objects are the visual components of the environment which are created in VRML.

This simulator also performs real time marking which enables trainees to make landmarks on the head immediately by clicking on the head. This capability together with other functions is achieved by combining VRML and Java EAI software.

2. Model Interaction

In the surgical procedure of Percutaneous Rhizotomy, it is an important that objects respond in real-time to user interactions. This simulation is performed by the user by using 2D mouse device. The interaction here includes the collision detection and the movements of the objects.

- **Collision Detection**

  When inserting the needle, it is useful to distinguish collision with the intended target (foramen ovale), and collision with the skull (which is incorrect). There should be different feedback (e.g. force feedback or visual feedback) when the needle touches the different objects. However, VRML is not able to detect collision between objects, only collision between object and the viewer's avatar. To make use of the collision detection provided by VRML browser for its avatars, this simulator binds a viewpoint (*Local View*) to the tip of the needle. In the separated browser of the *Local View*, this needle acts as an avatar of the VRML browser. When the needle moves in the *Global View* which is controlled by the *Control View*, this *Local View* is moved accordingly. When the needle hits the skull, the Local View has a collision as the avatar collides with the scene. The signal is then passed to the *Global View* so that the needle is bumped back in this browser.

- **Movements of the objects**

  According to the requirement of this procedure, the movements of the needle are limited to three degrees of freedom: rotations around two axes and translation along one axis. Those movements are controlled in the *Control View* and passed to the *Global View* and *Local View*. The *transformation* between the different coordinates and other mathematics tool such as *Quaternions* are performed in Java EAI. They control the continuity movements in both the *Global View* and the *Local View*.

References

[1] Y. Akatsuka, T. Shibasaki, A. Saito, A.Kosaka, H. Matsuzaki, T. Asano, and Y. Furuhashi. *Navigation System for Neurosurgery with PC Platform*. In Proceedings of Medicine Meets Virtual Reality 2000, pages 10-16. ISO Press, 2000.

[2] F. S. Azar, D. N. Metaxas, and M. D. Schnall. *A Finite Element Model of the Breast for Predicting Mechanical Deformations during Biopsy Procedures*. IEEE Workshop on Mathematical Methods in biomedical Image Analysis, 2000.

[3] M. Baker. Binary Space Partitioning (BSP) Tree website. http://www.martinb.com/ threed/solidmodel/spatialdecomposition/bsp.htm. Visited June 2002.

[4] D. Baraff and A. Witkin. *Large Steps in Cloth Simulation*. Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH, pages 43-54, 1998.

[5] Y. Bar-Cohen, C. Mavroidis, M. Bouzit, B. Dolgin, D. Harm, G. Kopchok, and R. White. *Virtual reality robotic telesurgery simulations using MEMICA haptic system*. SPIE Smart Structures and Materials, Electroactive polymer actuators and devices, pages 357-363, 2001.

[6] C. Basdogan, C. Ho and M. A. Srinivasan. *Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration*. IEEE/ASME Transactions on Mechatronics, 6(3), 2001.

[7] T. Berlage. *Augmented Reality for Diagnosis based on Ultrasound Images.* In Proceedings of the First Joint Conference of CVRMed II and MRCAS III, Springer Verlag, LNCS 1205, pages 253-262, 1997.

[8] D. Bielser and M. H. Gross. *Interactive Simulation of Surgical Cuts.* In Proceedings of Pacific Graphics 2000, IEEE Computer Society Press, pages 116-125, 2000.

[9] Blaxxun Contact website. http://www.blaxxun.com/. Visited September 2002.

[10] R. Boian, A.Sharma, C. Han, A. Merians, G. Burdea, S. Adamovich, M. Recce, M. Tremaine, and H. Poizner. *Virtual Reality-Based Post-Stroke Hand Rehabilitation*. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 64-70. ISO Press, 2002.

[11] P. Borrel and A. Rappoport. *Simple Constrained Deformations for Geometric Modeling and Interactive Design*. ACM Transactions on Graphics, 13(2), pages 137 – 155, April 1994.

[12] D. Bowman, J. Gabbard, and D. Hix. *A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods.* Presence: Teleoperators and Virtual Environments, 11(4):404-424, 2002.

[13] D. A. Bowman, D. B. Johnson, and L. F. Hodges. *Testbed Evaluation of Virtual Environment Interaction Techniques.* Presence: Teleoperators and Virtual Environments, 10(1):75-95, 2001.

[14] K. Brodlie, N. El-Khalili, and Y. Li, *Using Web-based Computer Graphics to Teach Surgery*. Computers and Graphics, 24(1):157-162, 2000.

[15] M. Bro-Nielsen. *Medical Image Registration and Surgery Simulation*. PhD. Thesis, 1996. Available at: http://www.imm.dtu.dk/~bro/phd.html.

[16] M. Bro-Nielsen. *Fast Finite Elements For Surgery Simulation*. Proceedings of Medicine Meets Virtual Reality 5 (MMVR '97), pages 395-400, 1997a.

[17] M. Bro-Nielsen. *Simulation Techniques for Minimally Invasive Surgery*. Journal of Minimally Invasive Therapy & Allied Technologies (MITAT), 6:106-110, 1997b.

[18] M. Bro-Nielsen. *Finite Element Modeling in Surgery Simulation*. Proceedings of the IEEE. 86(3):490-503, 1998.

[19] M. Bro-Nielsen and S. Cotin. *Real-time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation*, Computer Graphics Forum, 15(3):57-66 (Eurographics'96), 1996.

[20] M. Bro-Nielsen, D. Helfrick, B. Glass, X. Zeng, and H. Connacher. *VR simulation of Abdominal Trauma Surgery*. In Proceedings of Medicine Meets Virtual Reality 6 (MMVR'6), pages 117-123. IOS Press, 1998.

[21] M. Bro-Nielsen, J. L. Tasto, R. Cunningham, and G. L. Merril. *PreOp^{TM} Endoscopic Simulator: A PC-based Immersive Training System for*

Bronchoscopy. In Proceedings of Medicine Meets Virtual Reality 7, pages 76-82. ISO Press, 1999.

[22] Brouwer, J. Ustin, L. Bentley, A. Sherman, N. Dhruv, and F. Tendick. *Measuring In Vivo Animal Soft Tissue Properties for Haptic Modeling in Surgical Simulation*. In Proceedings of Medicine Meets Virtual Reality 2001, pages 69-74. IOS Press, 2001.

[23] J. Brown, K. Montgomery, J. C. Latombe, and M. Stephanides. *A Microsurgery Simulation System*. Medical Image Computing and Computer-Assisted Interventions (MICCAI 2001), Utrecht, The Netherlands, 2001.

[24] J. Brown, S. Sorkin, C. Bruyns, J. C. Latombe, K. Montgomery, and M. Stephanides. *Real-Time Simulation of Deformable Objects: Tools and Application*. Computer Animation 2001, Seoul, Korea, 2001.

[25] P. M. Chapman and D. P. M. Wills. *Towards a Unified Physical Model for Virtual Environments*. Proceeding of Fourth UK VR-Sig Conference, pages 130-139, 1997.

[26] Y. Chen, Q. Zhu, and A. Kaufman. *Physically-based Animation of Volumetric Objects*. Proceedings of Computer Animation '98, pages 154-160, 1998.

[27] C. Chronaki, X. Zabulis, N. Stathiakis, E. Leisch, D. Katehakis, M. Tsiknakis, and S.C. Orphanoudakis. *A Web-Based Environment for Medical Collaboration in a Regional Health Care Network.* Proceedings of CAR'97, pages 656-661, Berlin, Germany, 1997.

[28] S. Coquillart. *Extended Free-Form Deformation : A Sculpturing Tool for 3D Geometric Modeling*. In Computer Graphics, 24(4):187-196, 1990.

[29] S. Coquillart and P. Jancene, *Animated Freeform Deformation: an Interactive Animation Technique*. Computer Graphics (SIGGRAPH'91), 25:23-26, 1991.

[30] Cortona website. http://www.parallelgraphics.com/. Visited September 2002.

[31] CosmoPlayer website. http://ca.com/cosmo/. Visited September 2002.

[32] S. Cotin, H. Delingette, and N. Ayache. *Real-time Elastic Deformations of Soft Tissues for Surgery Simulation*. IEEE Transactions On Visualization and Computer Graphics, 5(1):62-73, 1999.

[33] S. Cotin, H. Delingette, and N. Ayache. *A hybrid Elastic Model Allowing Real-Tme Cutting, Deformations and Force-Feedback for Surgery Training and Simulation*. In Visual Computer, 16(8):437-452, 2000.

[34] S. Cotin, H. Delingette, J.M. Clement, M. Bro-Nielsen, N. Ayache, and J. Marescaux. *Geometrical and Physical Representations for a Simulator of Hepatic Surgery*. In Proceedings of Medicine Meets Virtual Reality '96, pages 139-151, 1996.

[35] S. L. Dawson and J. A. Kaufman. *The Imperative for Medical Simulation*. Proceedings of the IEEE, 86(3): 467-483, 1998.

[36] L. Davis, Y. Ha, S. Frolich, G. Martin, C. Meyer, B. Pettitt, J. Norfleet, K. Lin, and J. Rolland. *Augmented Reality and Training for Airway Management Procedures*. In Proceedings of Medicine Meets Virtual Reality 2002, pages 121-126. ISO Press, 2002.

[37] H. Delingette. *Towards Realistic Soft Tissue Modeling in Medical Simulation*. Proceedings of the IEEE: Special Issue on Surgery Simulation, pages 512-523, 1998.

[38] P. Dev. *Imaging and Visualization in Medical Education*. IEEE Computer Graphics and Applications, pages 21-31, 1999.

[39] M. Downes, M. Cavusoglu, W. Gantert, L. Way, and F. Tendick. *Virtual Environments for Training Critical skills in laparoscopic surgery*. In Proceedings of Medicine Meets Virtual Reality'98, pages 316-322. IOS Press, 1998.

[40] N. El-Khalili. *Surgical Training on the World Wide Web*. Ph.D. Thesis, University of Leeds, UK, 1999.

[41] N. El-Khalili, K. Brodlie, and D. Kessel, *WebSTer: A Web-based Surgical Training System*. In Proceedings of Medicine Meets Virtual Reality 2000, pages 69-75. IOS Press, 2000.

[42] Y. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. New York, Springer Verlag, 1993.

[43] S. F. F. Gibson. *3D Chainmail: a Fast Algorithm for Deforming Volumetric Objects*. In Michael Cohen and David Zeltzer, editors, 1997 Symposium on Interactive 3D Graphics, pages 149-154. ACM SIGGRAPH, 1997.

[44] S. F. F. Gibson. *Using Linked Volumes to Model Object Collisions, Deformation, Cutting, Carving, and Joining*. IEEE Transactions on Visualization and Computer Graphics, 5(4), 1999.

[45] S. Gibson, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, A. Mor, S. Nakajima, H. Ohkami, R. Osborne, J. Samosky, and A. Sawada. *Volumetric Object Modelling for Surgical Simulation*. Medical Image Analysis, 2(2):121-32, 1998.

[46] S. F. F. Gibson and B. Mirtich. *A Survey of Deformable Modeling in Computer Graphics*. Technical Report TR-97-19, Mitsubishi Electric Research Laboratories, Cambridge, MA, 1997.

[47] S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima, H. Ohkami, R. Osborne, and A. Sawada. *Simulating Arthroscopic Knee Surgery using Volumetric Object Representations, Real-Time Volume Rendering and Haptic Feedback.* In Proceedings of CVRMed/MRCAS'97, March 1997, Grenoble, France. SPL Technical Report #45, posted April 1997.

[48] S. Gottschalk, M. C. Lin, and D. Manocha. *OBB-tree: A Hierarchical Structure for Rapid Interference Detection*. In Proceedings of SIGGRAPH '96, pages 171-180, 1996.

[49] R. W. Gunderson, S. J. Smith, and B. A. Abbott. *Applications of Virtual Reality Technology to Wheelchair Remote Steering Systems*. European Conference on Disability, Virtual Reality and Associated Technologies, Maidenhead Berkshire UK, pages 47-56, 1996.

[50] D. Hix, J. E. Swan II, J. L. Gabbard, M. McGee, J. Durbin, and T. King, *User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment*. Proceedings of IEEE Virtual Reality '99, Computer Society Press, pages 96-103, 1999.

[51] G. Hirota, R. Maheshwari, and M. Lin. *Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization.* Proceedings of ACM Solid Modeling, pages 234-245, 1999.

[52] W. M. Hsu, J. F. Hughes, and H. Kaufman. *Direct Manipulation of Free-form Eeformations*. Computer Graphics, 26(2):177-184, 1992.

[53] S. M. Hu, H. Zhang, C. L. Tai, and J. G. Sun. *Direct manipulation of FFD: Efficient Explicit Solutions and Decomposible Multiple Point Constraints*, *Visual Computer*, 17(6): 370-379, 2001.

[54] T. Hudson, M. Lin, J. Cohen, S. Gottschalk, and D. Manocha. *V-COLLIDE: Accelerated Collision Detection for VRML.* In the Proceedings of VRML'97, 1997.

[55] Immersion Corporation website: http://www.immersion.com. Visited May 2002.

[56] D. Inman, J. Peaks, K. Loge, and V. Chen. *Teaching Orthopedically Impaired Children to Drive Motorized Wheelchairs in Virtual Reality*. Proceedings of the 2nd Annual International Conference on Virtual Reality and Persons with disabilities, San Francisco, CA, pages 55-58, 1994.

[57] ISO 9126 website: http://www.isaca.org.za/Iso9126.htm. Visited May 2002.

[58] D. L. James and D. K. Pai. *ArtDefo, Accurate Real Time Deformable Objects*. In Proceedings of SIGGRAPH 99 Conference, Annual Conference Series, pages 65-72. ACM SIGGRAPH , Addison Wesley, August 1999.

[59] Java 3D website. http://java.sun.com/products/java-media/3D/. Visited May 2002.

[60] N. W. John and N. Phillips. *Surgical Simulators Using the WWW*. In Proceedings of Medicine Meets Virtual Reality 2000, pages 146-152. IOS Press, 2000.

[61] N. W. John, N. Phillips, R. Vawda, and J. Perrin. *A VRML Simulator for Ventricular Catheterisation.* Eurographics UK, Cambridge, pages 1-6, 1999.

[62] N. W. John and M. Riding. *Surgical Simulators on the World Wide Web – This Must Be The Way Forward?* In Proceedings of UKVRSIG, Salford, UK, pages 25-34, 1999.

[63] N. W. John, M. Riding, N. Phillips, S. Mackay, and L. Steineke et al.. *Web-based Surgical Educational Tools*. In Proceedings of Medicine Meets Virtual Reality 2001, pages 212-217. IOS Press, 2001.

[64] Joint Committee on Higher Surgical Training. *A Manual of Higher Surgical Training in the United Kingdom and Ireland.* Seventh Report, July 1999. Available at: http://www.rcsi.ie/postgraduate_surgery /surgeons_in_training/pink_book.pdf.

[65] F. A. Jolesz, W. E. Lorensen, H. Shinmoto, H. Atsumi, S. Nakajima, P. Kavanaugh, P. Saiviroonporn, S. E. Seltzer, S. G. Silverman, M. Phillips, and R. Kikinis. *Interactive Virtual Endoscopy*. AJR, 169(1229-1235), 1997 SPL Technical Report #42, 1997.

[66] M. Mastrangelo, J. Stich, J. Hoskins, W. Witzke, I. George, J. Garrison, M. Nichols, and A. E. Park. *Advancements in Immersive VR as a Tool for*

Preoperative Planning for Laparoscopic Surgery. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 274-279. ISO Press, 2002.

[67] C. Kaufmann and A. Liu. *Trauma Training: Virtual Reality Applications*. In Proceedings of Medicine Meets Virtual Reality 2001, pages 236-241. ISO Press, 2001.

[68] C. Kaufmann, S. Zakaluzny, and A. Liu. *First Steps in Eliminating the Need for Animals and Cadavers in Advanced Trauma Life Support®.* In Medical Image Computing and Computer-Assisted Intervention (MICCAI), Lecture Notes in Computer Science 1935, pages 618-623, 2000.

[69] E. Keeve, S. Girod, R. Kikinis, and B. Girod. *Deformable Modeling of Facial Tissue for Craniofacial Surgery Simulation*. Computer Aided Surgery, John Wiley & Sons Inc., New York, invited paper, 3(5): 228-238, 1998.

[70] J. Kettenbach, F.A. Jolesz, and R. Kikinis. *Surgical Planning Laboratory: a New Challenge for Radiology?* In Proceedings of Computer Assisted Radiology and Surgery, Berlin, Germany, pages 855-860, 1997.

[71] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, and Y. I. H. Parish. *Simulating Facial Surgery Using Finite Element Models*. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pages 421-428, 1996.

[72] R. M. Koch, S. H. M. Roth, M. H. Gross, A. P. Zimmermann, and H. F. Sailer. *A Framework for Facial Surgery Simulation*. Technical Report No. 326, Computer Science Department, ETH Zürich, 1999.

[73] U. Kühnapfel, H. K. Çakmak, and H. Maass. *Endoscopic Surgery Training using Virtual Reality and Deformable Tissue Simulation.* Computers & Graphics, 24:671-682, Elsevier, 2000.

[74] U. Kühnapfel, C. Kuhn, M. Hübner, H.G. Krumm, H. Maaß, and B. Neisius. *The Karlsruhe Endoscopic Surgery Trainer as an Example for Virtual Reality in Medical Education.* Minimally Invasive Therapy and Allied Technologies (MITAT) 6, pages 122-125, 1997.

[75] Y. Lee, D. Terzopoulos, and K. Waters. *Realistic Modeling for Facial Animation*. In Proceedings of SIGGRAPH 95 Conference, pages 55-62. ACM SIGGRAPH, Addison Wesley, 1995.

[76] M. Lin and S. Gottschalk. *Collision Detection between Geometric Models: A Survey.* In the Proceedings of IMA Conference on Mathematics of Surfaces, pages 602-608, 1998.

[77] Liu, C. Kaufmann, and Daigo Tanaka. *An Architecture for Simulating Needle-based Surgical Procedures*. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Lecture Notes in Computer Science 2208, pages 1137-1144, 2001.

[78] J. A. Lozano, M. Alcaniz, J. A. Gil, C. Moserrat, M. C. Juan, V. Grau, and H. Varvaro. *Virtual Food in Virtual Environments for the Treatment of Eating Disorders*. . In Proceedings of Medicine Meets Virtual Reality 02/10, pages 268-273. ISO Press, 2002.

[79] B. Marovic. *Visualization of 3D Fields and Medical Data using VRML*. Belgrade University Computing Center, University of Belgrade, Yugoslavia, July 16, 1997.

[80] S. Mason. *Web-based Virtual Reality for Surgical Training*. MSc thesis, University of Leeds, 2001.

[81] E. S. Mathews and S. J. Scrivani. *Percutaneous Stereotactic Radiofrequency Thermal Rhizotomy for the Treatment of Trigeminal neuralgia*. Mt Sinai J Med 2000; 67(4):288-299, 2000.

[82] Mayfield Clinic. *Percutaneous Stereotactic Radiofrequency Rhizotomy (PSR)*. http://www.mayfieldclinic.com/PE/PE-PSR.HTM. Visited May 2002.

[83] P. B. McBeth, A. J. Hodgson, A. G. Nagy, and K. Qayumi. *Quantitative Methodology of Evaluating Surgeon Performance in Laparoscopic Surgery*. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 280-286. ISO Press, 2002.

[84] C. Monserrat, V. Hernández, M. Alcañiz, M. C. Juan, and V. Grau. *Evaluation and Study of a new Deformable Model Based on Boundary Element Methods*. CARS'99 Computer Assisted Radiology and Surgery, Elsevier, editor, pages 860-864, 1999.

[85] C. Monserrat, M. C. Juan, M. Alcañiz, V. Grau ,and C. Knoll. *Deformation Simulation Algorithms of ElasticTtissues in "Real-time" based in Elasticity Theory*. In Proceedings of Medicine Meets Virtual Reality 1999, pages 21-22. ISO Press, 1999.

[86] C. Monserrat, U. Meier, M. C. Juan, M. Alcañiz, C. Knoll, V. Grau, F. Chinesta, and C. Duval. *A New Approach for The Real-time Simulation of Tissue Deformations*. Computer methods and programs in biomedicine,

2000. Available on web site: http://www.dsic.upv.es/~cmonserr/papers.html

[87] K. Montgomery. *Virtual Reality-based Surgical Trainer for Long-Duration Space Missions.* In Proceedings of Medicine Meets Virtual Reality 2001, pages 315-321. ISO Press, 2001.

[88] K. Montgomery, M. Stephanides, J. Brown, J. C. Latombe, and S. Schendel. A *Virtual Environment for Training in Microsurgery*. In SPIE-The Optical Engineering Society, v3639(1), pages 398-403, 1999.

[89] D. Nain, S. Haker, R. Kikinis, and W. E. L. Grimson. *An Interactive Virtual Endoscopy Tool.* Satellite Workshop at the Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2001), Utrecht, The Netherlands, pages 55-60, 2001.

[90] S. Nakajima, H. Atsumi, A. H. Bhalerao, F. A. Jolesz, R. Kikinis, T. Yoshimine, T. M. Moriarty, and P. E. Stieg. *Computer-Assisted Surgical Planning for Cerebrovascular Neurosurgery*. Neurosurgery, 41(2): 403-409, 1997.

[91] S. Nakajima, R. Kikinis, F. A. Jolesz, H. Atsumi, M. E. Leventon, W. E. L. Grimson, N. Hata, D. C. Metcalf, T. M.Moriarty, P. McL. Black, B. Garada, and E. Alexander III. *3D MRI Reconstruction for Surgical Planning and Guidance*. In Advanced Neurosurgical Navigation, pages 137-145, 1999.

[92] J. C. Nebel. *Soft Tissue Modelling from 3D Scanned Data*. In Deformable Avatars, pages 85-97, 2001.

[93] L. P. Nedel and D. Thalmann. *Real Time Muscle Deformations Using Mass-Spring Systems*. Proceedings of CGI '98, IEEE Computer Society Press, pages 156-165, 1998.

[94] H. Nienhuys and A. F. van der Stappen. *Combining Finite Element Deformation with Cutting for Surgery Simulations*. In A. de Sousa and J. Torres, editors, EuroGraphics Short Presentations, pages 43-52, 2000.

[95] H. Nienhuys and A. F. van der Stappen. *A Surgery Simulation Supporting Cuts and Finite Element Deformation*. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Lecture Notes in Computer Science 2208, Springer Verlag, 2001.

[96] M. M. North, S. M. North, and J. R. Coble. *Virtual Reality Therapy: An Effective Treatment for Psychological Disorders*. Virtual Reality in Neuro-Psycho-Physiology 1997. ISO Press, 1998.

[97] M. M. North, C. M. Schoeneman, and J. R. Mathis. *Virtual Reality Therapy: Case Study of Fear of Public Speaking*. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 318-320. ISO Press, 2002.

[98] L. Palmer. Collision Detection for Animation website. http://www.eimc.brad. ac.uk/research/coldet.html. Visited June 2002.

[99] S. Payandeh, A. J. Lomax, J. Dill, C. L. Mackenzie, and C. G. L. Cao. *On Defining Metrics for Assessing Laparoscopic Surgical Skills in a Virtual Training Environment*. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 334-340. ISO Press, 2002.

[100] D. P. Pertaub, M. Slater, and C. Barker. *An Experiment on Fear of Public Speaking in Virtual Reality*, Medicine Meets Virtual Reality 2001, pages 372-378. IOS Press, 2001.

[101] R. Phillips, M. Hafez, A. Mohsen, K. Sherman, J. Hewitt, I. Browbank, and K. Bouazza-Marouf. *Computer and Robotic Assisted Osteotomy Aroun the Knee*. In Proceedings of Medicine Meets Virtual Reality 2000, pages 265-271. ISO Press, 2000.

[102] Phobia lists. Treatment for Phobias website. http://www. phobialist.com/ treat.html. Visited May 2002.

[103] X. Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In Graphics Interface '95, Quebec, Canada, pages 17-19, 1995.

[104] Quaternion website: http://mathworld.wolfram.com/Quaternion.html. Visited May 2002.

[105] M. Riding and Nigel John. *Force-feedback in Web-based Surgical Simultors*. In Proceedings of Medicine Meets Virtual Reality 2001, pages 404-406. IOS Press, 2001.

[106] D. Rigamonti, H. Bryant, and O. Buston. *Implementing Anatomic VisualizeR Learning Modules in Anatomy Education.* In the Proceedings of The Third Visible Human Project Conference, 2000. Available at: http://www.nlm.nih.gov/research/visible/vhpconf2000/MAIN.HTM.

[107] S. H. M. Roth, M. H. Gross, S. Turello, and F. R. Carls: *A Bernstein-Bézier Based Approach to Soft Tissue Simulation.* In Proceedings of the Eurographics ‘98, Computer Graphics Forum, 17(3), pages 285-294, 1998.

[108] M. A. Schill, S. F. F. Gibson, H. J. Bender, and R. Manner. *Biomechanical Simulation of the Vitreous Humor in the Eye Using an Enhanced ChainMail Algorithm*. Proceedings Medical Image Computation and Computer Assisted Interventions, MICCAI'98, pages 679-687, October 1998.

[109] M. J. Schuemie, C. A. P. G. van der Mast, M. Krijn, and P. M. G. Emmelkamp. *Exploratory Design and Evaluation of a User Interface for Virtual Reality Exposure Therapy*. In Proceedings of Medicine Meets Virtual Reality 02/10, pages 468-474. ISO Press, 2002.

[110] T. Seamster, R. Redding, and G. Kaempf. *Applied Cognitive Task Analysis in Aviation*. Avebury Ashgate Publishing Ltd., England, 1997.

[111] T. W. Sederberg and S. R. Parry. *Free-form Deformation of Solid Geometric Models*. In Computer Graphics, 20(4):151-160, 1986.

[112] J. Sinnott and T. Howard. *SQUIDS: Interactive Deformation of Superquadrics for Model Matching in Virtual Environments. In* Proceedings of Eurographics UK 2000, pages 73-80, 2000.

[113] S. Smith, A. Wan, N. Taffinder, S. Read, R. Emery and A. Darzi. *Early Experience and Validation Work with Procedicus VA – the Prosolvia Virtual Reality Shoulder Arthroscopy Trainer*. In Proceedings of Medicine Meets Virtual Reality 7, pages 337-343. ISO Press, 1999.

[114] G. Székely and R. M Satava, *Where are we going? Virtual Reality in Medicine.* BMJ 1999, 319(1305), http://bmj.com/, 1999.

[115] F. Tendick, M. Downes, T. Goktekin, M. C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, M. Hegarty, and L. W. Way. *A Virtual Environment Testbed for Training Laparoscopic Surgical Skills*. In Presence, 9(3):236-255, 2000.

[116] D. Terzopoulos and D. Metaxas. *Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics*. IEEE Trans. Pattern Anal. Machine Intell., 3(PAMI-7):703-715, 1991.

[117] D. Terzopoulos and A. Witkin. *Physically Based Models with Rigid and Deformable Componen.* IEEE Computer Graphics & Applications, pages 41- 51, 1988.

[118] J. M. Tew and J. M. Taha. *Percutaneous Rhizotomy in the Treatment of Intractable Facial Pain (trigeminal, glossopharyngeal, and vagal nerves)*. Operative neurosurgical techniques, pages 1469-1484, 1995.

[119] M. R. Thompson. *Virtual Sculpting*. MSc thesis, the University of Leeds, 1995.

[120] J. Torkington, S.G.T. Smith, B. Rees, and A. Darzi. *The Role of the Basic Surgical Skills Course in the Acquisition and Retention of Laparoscopic Skill*. Surgical Endoscopy DOI: 10.1007/s004640000183, 2001.

[121] Trigeminal Neuralgia (TN) in the University of Florida website. *The Surgical Treatment of Trigeminal Neuralgia and Hemifacial Spasm – A Guide for You and Your Family.* http://www.neurosurgery.ufl.edu /FacultyPage/TicBrochure.html. Visited May 2002.

[122] R.Turner and D.Thalmann. *The Elastic Surface Layer Model for Animated Character Construction*. In Proceedings of ComputerGraphics

International '93, Lausanne, Switzerland, Springer-Verlag, Tokyo, pages 399-412, 1993.

[123] UCI Medical Center in the University of California website. *Neurosurgery*. http://www. ucihealth.com/Neurosurgery/. Visited May 2002.

[124] UNC Gamma Research Group website. http://www.cs.unc.edu/~geom/ collide/ index.shtml. Visited June 2002.

[125] VR Presence website. http://www.vrweb.com/. Visited August 2002.

[126] P. A. Warrick and W. R. J. Funnell. *A VRML-Based Anatomical Visualization Tool for Medical Education.* IEEE Transations On Information Technology In Biomedicine, 2(2), pages 55-61, 1998.

[127] B. Watson, V. Spaulding, N. Walker, and W. Ribarsky. *Evaluation of the Effects of Frame Time Variation on VR Task Performance.* VRAIS'97, IEEE Virtual Reality Annual Symposium, pages 38-44, 1997.

[128] Web3D Web Site. www.web3d.org. Visited May 2002.

[129] WebSET website. http://www.hoise.com/vmwc/projects/webset/articles/ websetHome.html. Visited May 2002.

[130] R. W. Webster, D. I. Zimmerman, B. J. Mohler, M. G. Melkonian, and R. S. Haluck. *A Prototype Haptic Suturing Simulator*. In Proceedings of Medicine Meets Virtual Reality 9, pages 567-569. ISO Press, 2001.

[131] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick. *Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes*. In A. Chalmers and T. M. Rhyne, editors, Proceedings of Eurographics, pages 349-358, 2001.

[132] S. Zachow, E. Gladilin, H. C. Hege, and P. Deuflhard. *Finite-Element Simulation of Soft Tissue Deformation.* In Lemke, H.U. et al, editors, Computer Assisted Radiology and Surgery (CARS), Elsevier Science B.V., pages 23-28, 2000.

[133] J. Zeng, C. Kaplan, J. Xuan, I. A. Sesterhenn, J. H. Lynch, and S. K. Mun, *Optimizing Prostate Needle Biopsy through 3-D Simulation*, SPIE Medical Imaging, San Diego, 1998. Available at: http://www.simulation. georgetown.edu/miccai98.html.

[134] Y. Zhuang and J. F. Canny. *Real-Time Simulation of Physically Realistic Global Deformations*. IEEE Visualization Late Breaking Hot Topics Proceedings, 1999. Available at: http://www.cs.berkeley.edu/~jfc/papers/ grouped.html

[135] Y. Zhuang and J. Canny. *Haptic Interactions with Global Deformations*, IEEE Int. Conf. on Rob. & Autom. (ICRA), 2000. Available at: http://www.cs.berkeley.edu/~jfc/papers/ grouped.html.