Using Haptics in a Networked Immersive 3D Environment

By
Chris Gunn

THIS THESIS FORMS THE BASIS OF A SUPPLICATION FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT THE SCHOOL OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING OF THE UNIVERSITY OF WESTERN AUSTRALIA.

© Copyright 2007
by
Chris Gunn
I. Abstract

Haptic (force) feedback has had limited use outside laboratory environments due to the high purchase cost of the equipment. Potential users and their managers have no evidence that the benefits that it provides outweigh the costs involved. While the cost can be expected to ease with time, at present it is limiting the spread of the technology. While haptics enthusiasts may be convinced of its value, there is a lack of evidence of the techniques and the types of applications that can benefit from a haptic interface.

This thesis examines the utility that haptics provides to a computer user. It does this by following a series of publications written by the author on various aspects of the subject and adds to these the results of some, as-yet unpublished, experiments. The publications describe several immersive, 3D prototypes and applications that were developed by the author and his colleagues in the CSIRO\(^1\) Virtual Environments Laboratory in Canberra, Australia between 1999 and 2006. The work shows that haptic feedback can be successfully integrated into artistic, planning and teaching environments and in these cases it can enhance the user’s perception of the virtual environment being depicted. All the software for these applications was built upon a framework that originated from the laboratory and is described in some of the papers included in the thesis.

The networking of haptic applications is covered by several of the papers as well as analysed more closely in the text. The author has been able to create collaborative haptic applications that run successfully over much larger distances than were previously thought possible, overcoming some of the problems introduced by the inherent latency of the network. It should be noted, however, that the solutions detailed will only apply to interaction with certain types of virtual objects and do not provide a general solution to latency that will cover any scenario. Nonetheless, the techniques do satisfactorily cover many of the situations that are likely to be encountered when developing virtual reality simulation software.

\(^1\)CSIRO – Commonwealth Scientific and Industrial Research Organisation (Australia)
II. Contents

I. ABSTRACT ...........................................................................................................I
II. CONTENTS..........................................................................................................III
III. ACKNOWLEDGEMENTS .................................................................................IV
IV. STATEMENT OF CANDIDATE CONTRIBUTION ..............................................V
CHAPTER 1. INTRODUCTION ...............................................................................1
CHAPTER 2. RELATED WORK .............................................................................13
CHAPTER 3. SIMULATING REALITY .....................................................................30
CHAPTER 4. AUGMENTING REALITY .................................................................55
CHAPTER 5. EXTENDING REALITY ......................................................................63
CHAPTER 6. FORCE FIELDS ..................................................................................80
CHAPTER 7. HAPTIC GUIDANCE USER STUDY ................................................89
CHAPTER 8. TRANS-WORLD HAPTICS ...............................................................112
CHAPTER 9. LATENCY AND STABILITY .............................................................118
CHAPTER 10. SURGICAL TRAINING .................................................................144
CHAPTER 11. ACCOMMODATING A CLASS .......................................................162
CHAPTER 12. A CLINICAL STUDY ......................................................................184
CHAPTER 13. SCULPTING IN PAIRS ....................................................................196
CHAPTER 14. CONCLUSION ...............................................................................221
APPENDIX A : EXPERIMENT CONSENT AND INFORMATION FORM ..............249
APPENDIX B : SCULPTING EXPERIMENT RAW RESULTS .................................251
APPENDIX C : MINE PLANNING EXPERIMENT PLAN .......................................252
APPENDIX D : MINE PLANNING EXPERIMENT RAW RESULTS ........................255
APPENDIX E : RESIDUALS ..................................................................................257
APPENDIX F : ANALYSIS OF VARIATION RESULTS ...........................................260
APPENDIX G : ABILITY-TO-IGNORE-FEEDBACK RESULTS ............................261
APPENDIX H : ACRONYMS ...............................................................................262
III. Acknowledgements

I would like to acknowledge and thank the following people and organisations:

1. My supervisor, Assoc. Prof. Amitava Datta, of the University of Western Australia for advice, guidance and encouragement.

2. The Co-operative Research Centre for Advanced Computational Systems [2], who provided the infrastructure for the initial stages of this work.

3. The Centre for Networking Technologies for the Information Economy (CeNTIE) [48] which is supported by the Australian Government through the Advanced Networks Program (ANP) of the Department of Communications, Information Technology and the Arts, who provided the infrastructure for the later stages of this work.

4. Dr Warren Muller for advice on data collection and assistance with statistical analysis appearing in Chapter 7 and appendices E, F and G.

5. Dr Kevin Smith for advice and feedback on the thesis.

6. Colleagues at the Stanford University School of Medicine (supported by NLM/NIH Contract N01-LM-3-3512) and colleagues in the CSIRO ICT Centre Networking Laboratory for assistance in conducting international trials.

7. Prof. Leroy Heinrichs, of Stanford University, for advice on anatomy and clinical aspects of teaching the cholecystectomy procedure.

8. My partner, Anne Baynes for her encouragement, patience and assistance with proof reading, as well as my children, Josie and Laura, for leaving me in peace when necessary.

9. CSIRO (Commonwealth Scientific and Industrial Research Organisation) for support.
IV. Statement of Candidate Contribution

This thesis mainly comprises 10 publications covering a related sequence of research performed between 1999 and 2006. I am the sole author of 3 of the papers, the primary author of 5 more of the papers and a secondary author of 2 papers. My contribution to the papers is detailed below. The papers are not presented in chronological order, but have been ordered according to theme. The thesis also contains a chapter (Chapter 7) on an experiment performed as part of my PhD research. This experiment was entirely my own work, but Dr Warren Muller contributed greatly to the statistical analysis of the results.

The papers:

**Virtual Artisan’s Workbench**
Gunn, C. Simulation Technology and Training Conference, Sydney, 2000

**Haptic Collaboration with Augmented Reality**

**Experiments in the Haptic Rendering of Constraints: Guiding the User**
Gunn, C., Marando, P. Simulation Technology and Training Conference, Melbourne, 1999

**Using Force Fields as a User Interface Device**
Trans-World Haptic Collaboration

Combating Latency In Haptic Collaborative Virtual Environments

Using Collaborative Haptics in Remote Surgical Training

A Remote Interactive Master Class in Surgery

A Networked Haptic Virtual Environment for Teaching Temporal Bone Surgery

Collaborative Virtual Sculpting with Haptic Feedback
Gunn, C. Virtual Reality Journal, Springer UK
Special Issue: Collaborative Virtual Environments for Creative People, 2006, pp 73-83
Chapter 1.

Introduction

The thesis statement is that…

*Artificial force feedback can enhance a user’s skill in a virtual environment and that such haptic applications can be developed, which allow users to touch, interact and generally collaborate together over distances spanning the world.*

The term *haptic* comes from the Greek word, “haptesthai”, to touch [100]. It encompasses both tactile and kinaesthetic effects. In 1968 Gibson described haptics as “the sensibility of the individual to the world adjacent to his body by the use of his body” [74].

This thesis addresses the use of current haptic technology to enhance a computer user’s capabilities when dealing with three dimensional environments. Haptics requires the application of some relatively complex hardware and software combinations and is often seen as being pertinent only to specialised applications. The thesis states that there are substantial benefits provided by adding haptic feedback to appropriate applications and that these benefits outweigh the complexity in setting such systems up. In particular, teaching systems which involve experiential components can be developed and these may involve multiple users, completely or partially immersed together in the virtual environment, interacting across a network. ‘Hapticising’ such environments allows the users to manipulate objects and even each other in more natural ways, freeing their cognitive processes to address the learning task at hand.
1.1 Background

Over the last three decades considerable progress has been achieved in the area of graphics computing, mainly due to the growth of processor speed and memory capacity. The explosion of multimedia and the use of video and image-based communications over the Internet has been one result of this development. Real time access and sharing of powerful realistic, but simulated, visuals are now within our capacity, enabling truly interactive multi-participant, multimodal and multimedia communication. Stereoscopic vision techniques have allowed three dimensional perception to be introduced into computer simulations. It is said that a picture is worth a thousand words. It could be argued that a three dimensional model should be worth many times more, especially if it can be handled and manipulated [242]. Virtual environment applications in areas of entertainment, design, geology and medicine exist but, using only sight and sound, they leave us without the ability to feel weight, surface texture, surface compliance or temperature. In recent years these capabilities have been enhanced by the development of technologies that allow the sense of touch to be incorporated along with sight and sound.

Touch has historically been considered a highly significant sense [176] [187]. It is a general term which usually refers to the tactile sensation of contact on the human skin. The study of all aspects of touch lags far behind that of other sensory modalities, such as sight and sound. One reason for this is that it can involve many closely related mechanisms, such as heat, texture, hardness, weight, shape and pressure [198]. The field is inherently multidisciplinary, involving the areas of robotics, experimental psychology, biology, computer science, systems and control, and others. The fact that most experiences of touch occur to a single individual, as opposed to a group, also explains why systems which produce sight and sound have expanded into widespread use, while those of touch are still in the experimental stage. Since a visual or auditory display can be experienced by a number of people at once, the expense of any technology involved is more economically viable. Visual displays can accommodate an audience from one up to thousands and from a distance of a few centimetres to hundreds of metres. The same applies to audio displays. Because touch requires some form of apparatus to be in contact with the user’s body, simultaneous interaction with large groups of people is currently impractical.
The term “touch” is often associated with the sensations of skin contact with objects, and sometimes referred to as tactile sensation. The tactile sensation includes components of temperature, vibration, slip, roughness and pressure [176]. It tells us about both the surface textures and geometry of objects. However, the term “touch” is also used to include the active exploration of the environment with the human body – typically the hand and arm – i.e. “to touch something”. The touch referred to in this case refers to the relationship between the hand’s active movement and any resistive force that is encountered by contact with any objects in the scene. To touch something implies moving the hand or finger into contact with the object and to experience the sensations associated with that contact. The contact is meaningful only in the context of the movement and the cognitive awareness of the objects in the environment that caused the contact [74]. As well as the tactile sensations on the skin, such a contact also stimulates the receptors in the muscles, tendons and joints of the arm. Lederman and Klatzky [128] investigated the exploratory actions that humans use to haptically interpret surface features of an object. They found that textural exploration involved mainly lateral movements while curvature was detected with rounded movements. They state that “The haptic system is a neural system that uses inputs to mechanoreceptors that are embedded in skin, muscles, tendons and joints. Many living organisms use haptics to learn about the concrete world and its properties by means of purposive manual exploration.” [127] (page 1). This is an exploratory and manipulative form of touch and it is significant that there is a conscious, active participation by the human involved, as it is only the combination of this, with the receptor stimulation, that enables us to understand the shape, weight and form of the objects that we are touching. Supplying one user’s experience of touch artificially to another user, not undertaking the same manipulation, is unlikely, therefore to produce any meaningful sensation. This is discussed further in Chapter 6.

Active exploration and corresponding muscle/tendon reaction can be referred to as a kinaesthetic effect. Kinaesthetic information concerns physical forces applied to an object and returned from that object. It takes advantage of human proprioception – the awareness of body position through the muscle and tendon positions. Burdea [38] discusses the mechanism of how the body determines limb position and shows how proprioception allows you to know the position of your hand when you cannot see it.
Tactile sensations are often included under the general term of haptics. These sensations incorporate vibrations and surface textures and are detected by receptors close to the skin. The body’s detection mechanism has a bandwidth response of between 50 and 350 Hz. [38]. Kinaesthetics deals with contours, shapes and sensations like the resistance and weight of objects. Alternately, Te’eni et al. categorise skin sensations into tactile, which includes roughness and friction, and somatic, which includes changes perceived during static contact with the skin, such as temperature [228].

Tactile capabilities are most acute on the finger-pad, with a resolution of 0.15 mm. Kinaesthetic resolution is a measure of the minimum degree of motion that a human can detect. It is best measured as an angular displacement and is about 2 degrees for the finger and wrist [219]. While the study of both haptic and tactile feedback in computer systems is in its infancy, haptics is somewhat more advanced, possibly because the mechanisms required to produce it are less complex than those required for skin sensations. In its simplest form, the user needs to just hold a device that can move under the command of the computer.

1.2 Haptic Hardware

Haptic systems were initially developed around the need to tele-operate objects from a distance, typically because proximity to the objects was dangerous. Early systems were purely mechanical and involved chains, cables and levers, but by 1952 incorporated electronic components and then computing apparatus to provide wider capabilities and more control [76]. It is only relatively recently that haptic devices capable of providing a plausible simulation of touch have become available. While the cost of these is still reasonably high, prices are coming down and several competing products are appearing on the market. There are a wide range of haptic devices ranging from haptic mice to full arm exoskeletons. Current commercial products allow haptic feedback through desktop interfaces, such as the FEELIt™ mouse [104] or the PHANToM™ arm [144]. Tactile feedback at the fingertips can be delivered through vibro-tactile gloves such as the CyberTouch™ and force feedback can be delivered to the fingers with the CyberGrasp™ from Immersion Corporation [105]. Alternatively, pressure can be applied to points on the hand through inflatable pads [37]. The CyberGrasp™, is an exoskeleton device that fits over a 22 DOF CyberGlove™, providing per-finger force feedback. Exoskeleton devices consist of complex mechanisms that fit around the hand.
and control wrist and finger movement [121] [222]. Similar to the CyberGrasp is the Rutgers Master II [37] [36], which has an actuator platform mounted on the palm of the hand that gives force feedback to four fingers. Position tracking is done by the Polhmeus Fastrak™ [172]. It is possible for a high-bandwidth haptic device to portray texture as well as force through the device itself by providing high frequency vibrations [120] [166].

The most widely used haptic device is the PHANToM™ from SensAble Technologies [201]. Two PHANToM™ Premium 1.5s and a PHANToM™ Desktop haptic device were used for the work covered by this thesis. The PHANToM™ Premium is a pen-like device that has six degrees of freedom and renders three degrees of force information. Its use in interacting with virtual environments is described by its co-inventor, Ken Salisbury, in [195]. It can be held in the hand and provides three dimensions of force at the tip of the pen (or stylus). It can track the position and orientation of the tool within a workspace of 38 cm wide, 26 cm high and 19 cm deep. The maximum force exerted is 8.5 Newtons. The PHANToM™ Desktop model has a workspace of 16 cm wide, 12 cm high and 12 cm deep with a maximum force of 7.9 Newtons [201]. A more advanced model provides three dimensions of torque as well as three dimensions of force.

The PHANToM™ has a precise mechanism that can provide a convincing haptic sensation. In a typical configuration, every millisecond the computer reads the joint encoders to determine the position of the tip of the stylus. It then compares this position to the surfaces of any virtual objects in the scene. If the stylus is not touching a virtual object, no voltage is applied to the motors and no force is returned to the user. If the system detects a collision between the stylus tip and one of the virtual objects, it applies a voltage to the motors to apply a force to the user’s hand. This force, being a vector, can be designed to act in the virtual surface normal direction. Variations on this can be used to simulate friction and rough surfaces. Given sufficient power, the motors can prevent the virtual tool (and user’s hand) from penetrating the surface.

Exoskeleton devices incorporate a mechanical apparatus that is clamped to a user’s hand or arm like an external skeleton, and can exert forces on them. They operate under a similar principle, but with different mechanical actuation systems for force generation. They are used to simulate the grasping of virtual objects with the hand. As opposed to tactile feedback, force feedback can actively prevent a user from moving into a restricted simulation space. A haptic device should ideally be grounded onto a solid
base that has comparable or greater mass than the user. This is necessary to allow the
system to provide rigidity of motion, for example to halt the user’s hand when it comes
into contact with a solid virtual object.

Glove based haptic systems have the advantage of introducing force reactions into
grasping gestures, but are often difficult to calibrate and in most cases cannot provide
total rigidity of motion [37]. These devices, such as ungrounded exoskeletons, may
provide a finger flex sensation when colliding with a virtual object, but if there is no
grounding base to the equipment, there is nothing to stop the user from passing their
hand right through the object. The user may still perceive a momentary haptic effect
when passing the hand through an object, but this effect will not bear much resemblance
to the contact experienced when their hand contacts a real object. Nevertheless, such a
synthetic haptic sensation may still have a use, if the user can be trained to recognise the
sensation as an indication of a surface collision.

More realism can be introduced by combining a grounded system with a glove-based
exoskeleton. In these situations the exoskeleton glove is joined to a robotic arm device
attached to a grounded base. However, such a configuration introduces a safety issue.
Exoskeleton devices require the user’s hand to be fixed into the glove. A combined
system connects the glove to the haptic device. Any such system is effectively binding a
user’s arm into a computer-controlled robotic device and any software or hardware bugs
could potentially cause an injury to the user. It would be very difficult to create a system
that was sufficiently reliable to pass safety regulations.

1.3 Haptic Challenges

The application of haptics can be divided into two branches: teleoperation and virtual
reality. In teleoperation, a human operator is controlling a robotic device and receiving
forces based on the contacts the robot is making with real objects. The challenge in this
field is in detecting the contact forces without hindering the manipulation task and also
in overcoming stability issues associated with network latency if the operator and robot
are a long distance apart. The challenge in the use of haptics for virtual reality relates to
the accurate simulation of real world substances that appear both visually and haptically
realistic. Computationally efficient models and algorithms that match human perceptual
capabilities in accuracy and resolution need to be developed. The challenge here is to
create algorithms that are fast enough to be able to calculate the physical deformations
and other physical behaviour involved in object collisions, within the limited processing time available between successive haptic update cycles. The accurate simulation of haptic response is hampered by the complexity of the physics involved. It involves many closely related mechanisms, such as heat, texture, hardness, weight, shape and pressure [198]. Sometimes virtual reality constructs, such as virtual fixtures which provide artificial forces to influence a user's actions, can be used to assist the user to manipulate and interact with three-dimensional virtual objects, ibid.

Haptic interfaces create a bi-directional energy exchange between the user and the computer (unlike visual feedback which is unidirectional). This produces much stricter bandwidth limitations on the system to avoid instability. Devices typically need to be refreshed at rates in the order of 1000 Hz to avoid noticeable vibrations [15] [115]. This restriction typically limits the complexity of the simulated models, often resulting in a less than realistic appearance. As higher powered computers become available, the complexity can be increased in line with the hardware, but also advances in the optimization of the simulation software can reap performance rewards. Another avenue to extend the available processing hardware to accommodate more complex models is to incorporate approximations to reality where appropriate. If carefully chosen, these approximations can go un-noticed by the user, but can produce a vastly improved effect in areas that are noticeable. In 2004, Hayward suggested that “interactive presentation of data does not have to imitate reality, not even remotely. Being suggestive is what matters the most” [93] (page 4). This path has been chosen for the “pseudo physics” work addressed by this thesis (see Chapter 9).

There are also considerable challenges in the development of improved hardware used for haptic feedback. Resolution and bandwidths are still below that required to consistently produce suspension of disbelief in users, when touching and interacting with surfaces. Joint friction and component inertia need to be reduced. Multi finger and body devices are cumbersome, complex to calibrate and difficult to use.

Most haptic rendering is currently surface based. There has been some work on volumetric haptic rendering [219] [150], but the update rate requirements provide a challenging environment for the necessary calculations. Mazzella et al. provide an interesting approach to this problem by periodically storing force values based on a deformation simulation running at a low update rate, and interpolating these at the
higher haptic rates [146]. Bruyns and Montgomery [35] extended this to enable probing, piercing, ablating and cauterising of virtual tissues.

1.4 Scope and Themes of the Study

This study covers the application of haptics in a number of scenarios. It explores the problems of using haptics in collaborative applications and presents some solutions. The study does not claim to provide a catalogue of input devices and how they work. Other sources provide excellent reviews of haptic technologies [93] [38]. It also does not claim to have answers to physically accurate simulation of human tissue deformation. The chapters on surgical simulation describe training applications that are designed to teach the procedural aspects of surgery. The simulation of tissue deformation, diathermy and cutting, as well as temporal bone drilling is not an attempt to reproduce the physically correct behaviour of the material in contact with surgical instruments. It has sufficient subjective realism to allow trainees to accomplish each stage of the procedure in a similar manner to actual surgery, thus providing the affordance necessary to free their attention for the process of learning the components of the procedure, the sequence of tasks and the risks and landmarks to look out for.

My work was carried out using the SensAble Technologies PHANToM 1.5 haptic device. This is a three degree of force device (3DOF) that has six degrees of freedom. This means that it can be manipulated within three position axes, (x, y, z) and 3 orientation axes, (roll, pitch, yaw). However, forces are only returned via the x, y and z axes. A consequence of the lack of force on the roll, pitch and yaw axes is that no torque forces can be represented. In simulation environments, torques may be needed to represent accurate force behaviour for any twisting motion, such as a screw-driver in a slot or a scalpel embedded in tissue. Torque is also necessary to represent inter-object collision with irregularly shaped objects or when friction is involved. Nevertheless, 3DOF haptic devices are still very useful for many simulation situations and can also provide an acceptable approximation for these more complex collisions. Six degree of force haptic devices are available [232] [201] [50] [64] [54]. Typically, these incorporate motors onto the orientation mechanism, but these are not investigated within the work covered by this thesis.
1.5 Organisation of the Thesis

This thesis is presented as a series of works published between 1999 and 2006. These papers have appeared in refereed conferences and journals as full papers, short papers and poster papers. Chapters 3-6 and 8-13 in this thesis each contain one of these papers, along with an introduction section before the paper and a discussion section afterwards. The intention of these is to fill in any gaps in the papers due to page limitations, to add any related new information that has come to light after publication and to discuss other possibilities and directions that the work could take.

The study covers my investigations into the use of haptics in virtual environments and the extension of those environments to operate across a network. Since different components of this work overlapped in time and paper publication dates depended on the particular conferences and journals being targeted, those publication dates do not strictly match the logical progression of the work. For this reason, the papers do not appear in this thesis in order of their publication dates, but have, instead, been ordered along a logical sequence according to their content.

Chapter 2: Related Work describes related work in haptic hardware, haptic surfaces, constraints, feedback methods, user interaction and haptic collaboration.

Chapter 3, Simulating Reality, describes an artistic environment where users can touch and feel the virtual objects they are creating.

Chapter 4, Augmenting Reality, presents an alternative hardware environment that allows virtual touchable objects to be mixed with live video.

Chapter 5, Extending Reality, discusses the concept of adding haptic effects which may not exist in the real world.

Chapter 6, Force Fields, investigates these ‘extra-real’ effects in more depth.

Chapter 7, Haptic Guidance User Study, reports on three quantitative experiments into the degree of assistance that force feedback can provide.

Chapter 8, Trans-World Haptics, is an introduction to a world-first haptic application that allows haptic collaboration from opposite sides of the Earth.

Chapter 9, Latency and Stability, details the experiments and algorithm development that enabled this long-distance collaboration to succeed. There are particular problems
concerning the physical stability of the equipment and logical consistency of the virtual objects that are addressed.

**Chapter 10, Surgical Training**, describes the application of the technology to the training of surgeons.

**Chapter 11, Accommodating a Class**, explains one method of incorporating more than one student into such a teaching session.

**Chapter 12, A Clinical Study**, describes the measurement of the benefit that the surgical training environment provides.

**Chapter 13, Sculpting in Pairs**, presents a completely different use of the collaborative haptic environment, to demonstrate its generality.

The thesis statement is addressed in **Chapter 14, Conclusion**.

The nature of this thesis results in the repetition of some information. This is because it contains a series of publications about a progression of research. Each publication needed to describe the context of the work before continuing with any new contribution. This is especially noticeable with the description of the hardware involved in the work, specifically the CSIRO Haptic Workbench.

This hardware configuration is sufficiently important to the understanding of the concepts being presented as the argument of each paper, that it would have detracted from each paper’s impact if it were omitted. Three dimensional interaction, especially involving stereo vision and touch, is difficult to convey in words. Often users are delighted and astonished when they first try a haptic application in the workbench environment. In the format of a conference paper, a demonstration is usually not practical, so a description of the hardware, accompanied by a diagram and photograph, is the next best option. It is needed to set the scene in the reader’s mind so that they can more easily imagine the user interface and therefore understand the software issues being discussed. Such a description of the hardware is not necessarily required for conventional, windows-based applications because of the audience’s familiarity with them.

In 1999-2002, when the first few papers in this series were written, the immersive, mirrored, co-located hardware configuration was reasonably new and novel. In recent years, this type of hardware has become more common in virtual reality applications,
with several commercial versions available. For this reason, I considered removing these sections from most of the included papers, but after some consideration, I decided that for reasons of completeness, the papers should be left intact, as originally published. Once the reader is satisfied that they understand the hardware configuration, it may be convenient to skip further references to hardware setup within successive papers, as the hardware design did not change significantly during the period of the research. The one exception to this is contained in the work on collaborative applications, which necessitated the addition of video conferencing equipment into the hardware configuration. This first appears in Chapter 10.

All papers were limited in length to some degree by the publishing organisation. Consequently some interesting facts were, of necessity, omitted. This particularly relates to “Trans-World Haptic Collaboration” presented in the Sketches and Applications section at SIGGRAPH 2003 (Chapter 8), which was limited to one page. For this reason another paper relating partly to the same work, “Using Collaborative Haptics in Remote Surgical Training”, presented at WorldHaptics in 2005 (Chapter 10), is also included in the thesis. The reader should treat the former paper as an introduction and the second as a more detailed coverage.

Chapter 7, “Haptic Guidance User Study” differs from the other chapters, in that it covers new work not relating to any prior publication. It describes experiments I performed as part of the PhD research covered by this thesis, to quantify the improvement in user skill that can be gained by adding haptic feedback to a three dimensional task.

The thesis concludes with a conclusions, bibliography and appendices. Note that each individual paper originally contained its own shorter bibliography. For the sake of clarity, these citations have been moved to a single thesis bibliography at the end of the work. As well as these citations, the thesis bibliography includes a wider ranging literature search on the subject, reported in Chapter 2.

1.6 Text Style

*Italics* are used in mathematical equations, for emphasis and for terms associated with a particular discipline, and that are not in common use (e.g. *decline* used in mining).

Equations are shown in grey background boxes, as in \[a = b + c\]
Vector variables in equations are shown with a superscript bar, as in $\vec{v} = \vec{v}_1 + \vec{v}_2$.

This work was undertaken in a virtual environments laboratory, where I worked within a team, but where each individual undertook separate research tasks. Many of the included papers report on the team’s work, but with particular emphasis on the individual work of the first author. Consequently, “we” is often used in preference to “I” within the papers for personal pronoun references, regardless of whether the piece of work was an individual effort or done with others. However, explicit statements in the introductions to each chapter detail exactly what parts and what proportion of the work is my own.

Computer code is shown in courier font.
Chapter 2.

Related Work

Computing systems traditionally provide feedback to users in the form of a visual display (i.e. text and pictures on a screen), or sometimes an audio display (i.e. sounds through speakers). More recently, computing systems have ventured into the area of providing haptic feedback – i.e. feedback that involves some form of touch.

Haptic feedback was first used in the 1950s with the development of mechanical systems for the manipulation of hazardous nuclear materials [222] [76]. A system of rods, chains and cables allowed operators to remain protected from the radio-active material while having enough mechanical feedback to allow contact forces to be perceived. The operator would work on a master system and a slave system mimicked the actions of the master to operate on the material.

The first use of computer generated haptics occurred in the 1960s, when Goertz and his colleagues at Argonne National Laboratory in the US built robotic manipulators which incorporated force reflection back to the controller [77]. These were used to remotely handle dangerous nuclear fuel rods in the emerging nuclear power industry. Also in the 1960’s Rotchild and Mann used force feedback in artificial limbs for amputees [24]. However, at that time, computing power was too low to allow the computation of anything like realistic forces in real time.

By the 1970’s, the mechanical separation of the slave and master, via an electronic communications link permitted teleoperation techniques to be taken up [116] [17] [194]. This was also applied to under-sea and space exploration [119]. By this time it was possible to calculate the mechanical models of object behaviour with enough precision to allow a master haptic arm to interact with a simulated model of objects, as opposed to
objects in the real world. The sensation of touch could be re-created at a master arm by computing the interaction forces on the virtual model [159].

One of the earliest introductions of haptic feedback in a virtual reality context (as opposed to a teleoperation context) was undertaken in 1993 by Adachi et al. [3] [142]. Rather than calculating a single force directly from the interaction, they supplied an intermediate representation for a force model. Another early use of haptic feedback in virtual reality was the use of a master arm to manipulate virtual molecular models, such as in the GROPE project [34]. The addition of haptic feedback was found to increase users’ performance in the manipulation tasks. In 1997 Srinivasna and Basdogan coined the term “computer haptics” to describe the discipline of providing artificial haptic stimuli to a user (in the same way that computer graphics deals with generating visual images) [219]. “Haptic rendering” was defined by Salisbury et al. as “the process of computing and generating forces in response to user interactions with virtual objects.” [194] (page 123).

2.1 Hardware

There have been various combinations of graphic displays with haptic hardware. In 1996 Posten et al. first developed a mirrored display device enabling co-located visual and proprioceptive interaction with a virtual 3D model [173]. This involved stereoscopic viewing of the model located beneath a mirror, allowing the user’s hands to be located in the same work area without occluding the virtual objects. In 1999, Stevenson et al. added haptic feedback to this arrangement for increased realism [221], and the scaling issues of this are discussed by Bogsanyi and Krumm-Heller in [25]. In 2000, Bouguila investigated haptics and stereopsis (3D vision) cues in manipulation of virtual objects and found that they both improved the time needed to perceive depths [28]. The importance of stereopsis is demonstrated by Galyean and Hughes [73] who discovered difficulties when no stereoscopic viewing was available.

Haptic devices can range in complexity from haptic mice [154] [7] to shoulder mounted whole arm configurations [98] [121]. The first widely available commercial haptics device was the PHANToM haptic interface that was developed in 1993 [144] [195] [201]. The PHANToM™ is a haptic device that is held lightly in the hand and can be manipulated in the similar way to common real-world tools like a screw-driver, pen, probe or knife. This allows the user to perform familiar hand and wrist actions and
perceive a matching representation in the virtual scene. Burdea [39] describes this along with the Rutgers force feedback glove and other devices which have been developed to allow whole hand interaction [105]. He provides some history on the development of haptics for teleoperation, a precursor to the use of haptics within virtual environments, such as is addressed in this thesis. In 2005 Méndez et al. [155] described the proactive desk, a magnetic desk with a projected display onto it. Haptics are created through a magnetic stylus used on the surface of the desk. This is an interesting way to provide forces to the hand without the constraint of attachments, but only works in two dimensions. Hayward [93] provides a comprehensive description of currently available haptic devices. He categorises them into impedance devices and admittance devices. Impedance devices are those in which the user changes the position of the tool and the system may respond with a generated force if necessary. Admittance devices are those where the tool measures any force applied by the user’s hand, and then uses its motors to reposition the tool if necessary [140] [230]. Wu et al. describe admittance device control as “scaling an input force and outputting a displacement or velocity” [250]. Admittance devices have not penetrated the market as well as impedance devices because of the complexity of accurately detecting the input forces, but they have the potential to simulate stiffer surfaces.

Haptic devices can alternately be divided into active and passive categories as described by Rosenberg et al. [189] in 1996. Active force feedback controllers apply forces to the user by adding energy into the human-machine system. Passive force feedback controllers apply forces to the user by removing energy from the human-machine system. They found that both styles improved user performance in a user interface widget manipulation task.

In the book, “3D User Interfaces” Bowman et al. [30] reported in 2004 that multiple DOF (degree of freedom) devices with integrated control of all dimensions are usually the best for 3D object manipulation. They explain that “Integrated control allows the user to control the 3D interface using natural, well-co-ordinated movements, similar to real world manipulation.” In 1992, Shimoga [209] showed that the ability of the hand to exert forces has a 5-10 Hz bandwidth, but can sense changes in forces at a bandwidth of 20-30 Hz. It also has a tactile bandwidth of 0-400 Hz. Earlier, in 1987, Smith and Smith found that the sensory dimensions needed to correspond to each other [213]. For example, performance degrades if the visual feedback conflicts with the kinaesthetic
feedback. The importance of matching the different sensory cues was verified by Bresciani in 2005, who reported that the human central nervous system tends to “bind” visual, auditory and tactile stimuli, that seem to originate from the same event, into one logical signal [31]. As noted by Loomis in 1992 [137], successful interaction with a virtual or tele-presence environment hinges upon whether the operator experiences distal attribution, which helps observers to accept the sensory impressions of the real world as accurate representations of physical reality [75]. Lederman has performed extensive research into the neurological aspects of haptic interaction [128]. In this, she presents a cognitive scientist’s view on the design of teleoperation systems and together with Flanagan in 2001 [72], investigated haptic illusions that can be used to overcome hardware limitations, taking advantage of both the strengths and the limitations of human perception. In 2003, it was shown by Kuchenbecker et al. [124] that the tightness of a hand grip on a haptic device changes the bandwidth and stability of a closed loop haptic system. A firmer hold can provide some damping to the system, but can overpower the more subtle haptic effects. They suggest that a dynamic system could be developed that could vary system parameters depending on the user’s grip force.

In 2002, Aliakseyeu et al. compared four interaction devices - cubic mouse, electronic pad and stylus, horizontal and vertical screen and passive interface props – in work designed to investigate navigation within a 2D and 3D space [8]. Trials found that users quickly became familiar with the use of the devices and found the interface techniques useful and intuitive. They stated that:

Ideally, interactions with data should look and feel to the user like s/he is directly navigating and/or manipulating the data (at a certain level of abstraction), without the need to pay much attention to the interaction itself….As a general requirement for naturalness, the atomic actions required to perform the interaction tasks of interest should match the atomic actions provided by the interaction device.

The applications described in this thesis were also designed on these principles. The intention was to create an environment in which users manipulated the data directly, not through on-screen widgets.
2.2 Haptic Surfaces

There is substantial evidence to indicate that people prefer to both feel and see the objects that they use to interact with a computer in a work environment. For example, in 1994, Barrett et al. found that both touch typists and casual users had significantly better performance with, and more positive feelings toward, a standard keyboard than a flat keyboard that did not provide any haptic feedback as keys were pressed [12]. Bowman et al. state that "The human hand is a remarkable device: it allows us to manipulate physical objects quickly, precisely, and with little conscious attention" [30]. In the seminal work, What’s Real about Virtual Reality [33], Brooks states that “So much of our interaction with objects depends on how they feel—purely visual simulation misses realism by much too far. This is critical for some training tasks; we really do not know how important it is for design tasks.”

In 1992, Galyean and Hughes [73] found that without haptics, controlling hand position in 3D space is difficult. To overcome this they provided a false haptic effect with strips of elastic. Three years later, while working with 3D sketching, Deering et al. [63] discovered that they required correctional algorithms to remove hand tremor when a haptic resting surface was not available. Oakley et al. [162] found in 2000, that with a haptic interface, task errors were significantly reduced and subjective workload measurements showed that participants perceived many aspects of workload to be significantly less with haptics. In 2002, De Haan et al. [62] found that without an appropriate feedback it is difficult to freely probe data in 3D space. They found that passive haptic feedback, provided by a plexiglass pad was an excellent support during 3D probing.

Prior to 1989, Hannaford [91] investigated human factors in tele-operating a robotic manipulator for a peg-in-hole task. Real world direct manipulation was best, as might be expected. Teleoperation with force feedback resulted in 30% faster completion time, and reduced error rates by 60% compared to a user interface without it. The sum-of-squares of forces detected by the tele-operated robot interacting with the environment was reduced sevenfold. In 1990, Buttolo et al. [42] compared times for direct, virtual and remote manipulation of an object. They found that direct and virtual manipulation were similar. Remote manipulation took a greater time. In 1998, Miller and Zeleznik [148] found that providing haptic ridges and grooves between user interface buttons and menu items improved the speed of use with a haptic mouse over a non-haptic version.
They also found that the introduction of some non-physically based behaviour, such as using a haptic sphere size to control camera zoom, improved the efficiency of the interface [148] [149].

Creating a virtual haptic surface typically requires some form of approximation. Calculating the haptic response to the collision of a tool with a hard surface can be problematical within the response time required due to limitations in processing power. One method of producing the reaction force involved is the penalty-based system, described in 1988 by Moore and Wilhelms [152] as using the depth of penetration of the haptic tool within a solid as the determining measurement. This, however, can produce incorrect results with thin layers where the tool can exit on the other side. The constraint-based or ‘god-object’ method of rendering haptics avoids the problem by keeping a representation (or proxy) of the haptic tool tip which can diverge from the real tool tip position. It is always kept on the outside of any solid virtual surface, while the real haptic tip penetrates the surface. The distance between the visible tip and the real haptic tool tip is then used to calculate the force to be returned to the user. More detailed descriptions of this technique can be found in [254] [167] [191]. The haptic work covered by this thesis uses an implementation of this ‘god object’ method.

Several haptic rendering systems use a linear approximation to a surface, for haptic interaction, for the period (usually 30 ms or so) between graphics updates. In 1997, Ruspini et al. described one such system which also incorporated the ability to haptically render friction and texture as well as perform a haptic version of Phong shading [191]. Another system that accommodated deformable surfaces was described by Cavusoglu and Tendick in 2000 [47]. Mark et al. [142] used a separate computer to calculate the instantaneous linear approximation to the touch point on a surface and provide the haptic rendering while the main computer provided the graphics and simulation logic. The two communicated with TCP [220]. In the system used for my experiments, a similar method is employed, except that a dual processor machine is used, with one processor dedicated to haptics and the other to both graphics and logic. Communication is through shared memory. In 2001, Okamura et al. [166] investigated ways to improve the representation of a collision with a stiff surface by developing a physically accurate reproduction of the vibrations that occur when a tool collides with surfaces having various material properties. They measured users’ ability to determine the material type from experiencing the collisions by tapping the virtual material and
found that this method was effective and enhanced the realism of the interaction. In 2006, Kuchenbecker et al. reported that the addition of transient vibrations could enhance the perceived rigidity of a stiff virtual surface [123].

There appears to be a mismatch in the human interpretation of objects via different sensory modalities. Wuillemin et al. [251] investigated the perception of size between interface modalities in 2005. They found that the radius of virtual spheres was perceived by subjects to be different depending on whether they were presented with haptics or graphics representations of the sphere. Virtual haptic spheres were perceived to be larger than their virtual visual representation. Other experiments by Flanagan and Lederman in 2001 [72], investigated the illusion of portraying a vertical bump with a change in resistance to lateral movement. This is analogous to a car slowing down as it goes over a hill. Providing only a two dimensional haptics device, it was possible to provide the illusion of a bump in the third dimension.

2.3 Haptic Constraints

Early work on artificial haptics concentrated on the simulation of real world surfaces and objects. More recently there has been development of the use of forces to influence the user’s actions in some way, separately to simulating collisions with surfaces. These forces are often termed ‘haptic constraints’ or ‘virtual fixtures’. The most common form of constraint is the ‘snap-to’ effect\(^2\). This can be implemented with or without haptics. Without haptics it involves the cursor slipping into a particular point when it comes within range, regardless of the user’s actions. The haptic form involves a force towards the point, allowing the user to comply with it or resist. A non-haptic snap-drag interface was shown by Bier and also Butterworth et al. in the early nineties, to result in a reduced number of actions being required for a particular task [20] [41]. In 1991, Schmidt and Lee [200], stated that when haptics are used to help in a training task, the users can come to rely on the feedback, and therefore not be as efficient when performing the task without the assistance of the feedback. This is backed up by Kuang et al. who investigated the use of haptic constraints to provide guidance in training environments in 2004 [122]. They discovered that there was no improved learning for a haptics group over those given graphics clues only. Earlier work by Gillespie et al. [75]

\(^2\) a pull of the user’s virtual tool towards a single point in the scene
and Feygin et al. [69] also showed little improvement in skill-learning by haptic guidance except in the early, exploratory stages of learning. The issue is discussed more fully in Chapter 6. An interesting variation to using haptic guidance for training is to apply a force in the opposite direction to that which the student should move as investigated by Saga et al. [193] in 2005. The intention here is to force the student to strive against the force and therefore learn the required action more comprehensively through increased effort. The authors were able to demonstrate some improved learning with this method.

My work in this area is directed at using the haptic feedback to improve the skills of a worker while they are doing the task, not while they are training for it. The haptic constraints are intended as an ongoing aid, to advise and improve performance while in use, and are not intended to permanently improve a worker’s skill. As an analogy, a ruler helps in drawing a straight line [190]. It doesn’t train its user to draw straight lines without it. Takemoto et al. applied this technique in 2005 to assist crane operators when they are loading cargo onto ships [226]. They provided an automatic force to a crane operator’s controls when the cargo was getting close to an obstacle, allowing the operator to be aware of the obstacle and take appropriate action. They found that the appropriate force should be sufficient for the operator to be aware that some action was required, and it should indicate the direction that the action should take, but not be so powerful that the crane controls moved independently. This was to avoid the operator getting a surprise when the crane moved independently and out of his control. It also gave the operator the ultimate control to either comply with, or resist the automatically provided advice.

In 2002, Marayong et al. [141] investigated the influence of compliance on performance during human-machine cooperative manipulation. They particularly studied the effect on independent user actions – the ability to over-ride the haptic guidance. They found that a compliance of 60% (i.e. the user supplies 40% of the movement effort) produced the best performance in their experiments. A related approach was used by Nolin et al. in 2003 [158], who compared different methods of altering virtual fixture stiffness to determine the effect on performance. The experiments described in Chapter 7 detail my investigations into this issue by measuring the user’s ability to either comply with a guiding force, or over-ride it.
Payandeh and Stanisic [170] also studied the use of virtual fixtures for teleoperation and training, in 2002. They achieved a performance increase and reduced operator workload compared to ordinary manual control during a manipulation task. They describe the use of virtual fixtures for training but their experiments did not measure any training effect, nor do they report on any training improvement. Bowman et al. caution that there is a risk that introducing haptic constraints may make interaction more difficult and frustrating [30]. They also suggest that haptic constraints which reduce the number of degrees of freedom may simplify a task. For example a virtual object can be constrained to move only on the surface of a virtual plane, making its positioning easier within that plane. The difference between passive constraints which form boundaries, and active constraints, which move the user in a desired direction, is discussed by Komerska and Ware [117]. They also describe hard, 3D barriers in [118] and haptic navigation widgets in [119]. They state that "We have developed several principles to guide our effort, chief of which is the notion that haptic forces should be used to provide constraints on user tasks, rather than mimic physical object forces." [117] (page 270). However, my work (Chapter 7) indicates that haptics can usefully be applied in both of these ways. There has been work [132] into the use of Hidden Markov Models to detect a user’s intent in a 3D task, and to provide suitable haptic constraints to assist them in that task. The virtual fixture is then turned on or off according to the anticipated goal of the user.

An interesting use of haptic constraints was described by Sayers and Paul [197] in 1994, for the teleoperation of an undersea robot. The user performs small steps of the navigation operation in a virtual environment. The commands are transmitted to the slave robot beneath the sea. The constraints, or ‘synthetic fixtures’, are based on snapshots of the undersea environment sent back to the surface. The system is designed to overcome the problems of delay in transmission time.

Constraints have also been used in surgical systems to assist and guide a surgeon [61], [101]. In 1990, Park et al. [169] used virtual fixtures to guide a surgeon’s instrument during a robot-assisted coronary bypass operation. More recent work in 2005 [174], indicated that point attraction virtual fixtures could assist in a surgical task.

2.4 Comparing Haptic Feedback with Other Feedback Methods

Haptics is only one of a number of feedback methods that can be incorporated into a user interface. Comparative studies of haptic and other methods of user feedback agree
that haptic feedback has significant benefit in some areas. In 1995, Akamatsu et al. compared haptic (tactile), visual and auditory feedback in a target selection task [7]. A computer mouse was modified to add tactile feedback via a solenoid-driven pin projecting through a hole in the left mouse button. Then a target selection task was undertaken using the five different sensory feedback conditions of "normal", auditory, colour, tactile, and a combination of these. They found no differences in response times or error rates, but significant differences occurred in selection times (the time between the cursor entering the target and selecting the target), with tactile feedback providing the fastest. They argue that "tactile feedback allows subjects to use a wider area of the target and to select targets more quickly once the cursor is inside the target" (page 816).

In 1996, Mtinch and Stangenberg [154] experimented with a modified computer mouse by adding magnets to provide resistance to movement, plus a vibro-tactile pin. They found that this reduced positioning times by 10%. In 2002, Unger et al. [233] found that adding haptics to a virtual peg-in-a-hole task improved performance over a visual-only interface. The haptic feedback was provided by a six degree-of-force magnetic levitation device. A matching task using real pegs and holes resulted in the best performance, and this can be attributed to the difference in fidelity between the real world and that presented by virtual reality devices.

Although humans get approximately 70% of their sensory input from vision [243], several bodies of work [65] [205] [233] point to the ability of haptic feedback to improve both accuracy and speed in performing tasks, especially in three dimensions. Chapter 7 of this thesis explores this issue further. The 1995 work by Richard and Coiffet [184] showed that the presentation of redundant information, in the form of both haptic and graphic representation, increases performance and reduces error rates. A combination of haptic and visual modalities was found to be more effective than vision alone. In 2005, Grane and Bengtsson [79] found that mental workload was less with a haptic-plus-graphic interface, than with graphic only or haptic only. The work in Chapter 4 of this thesis describes the ‘hapticisation’ of a formerly visual-only environment, ARToolkit, a graphical system for registering virtual models within the real world via image recognition of a fiducial marker [22] [113] [114].

Chapter 7 contains a report on an experiment which also compares performance under different combinations of haptic, graphic and auditory feedback. However, this differs from the 1995 study by Akamatsu et al. [7] in that the haptic feedback contains
directional information (a force as opposed to a tactile stimulation). This is significant in that the feedback both alerts and advises the user on an action, whereas the tactile mouse only alerts the user. Other work in this area has discovered that haptics can improve times but not accuracy [7]. The provision of stereo depth cues was found to improve both speed and accuracy. My experiment reports an improvement in both speed and accuracy with haptic feedback and stereo vision, over a stereo-vision-only interface.

In 2000, Wall and Harwin [241] investigated the effect of haptics on user response in a Fitts’s Law experiment. They discuss ‘ballistic’ and ‘non-ballistic’ motion and define ‘ballistic’ as a rapid, involuntary movement which is motor programmed, where visual feedback is not possible. They found that Fitts's law is only applicable to visually controlled movements. They use an interesting analogy to illustrate this; they compared the effect of using force feedback to a drum roll, where a drum strike frequency of up to 40 Hz can be achieved despite the human bandwidth for limb motion being only about 10 Hz. This mechanism uses the haptic bounce of the stick on the drum and the human's ability to control this with a varying applied pressure.

The 2005 work by Tavakoli [227] investigated the use of a visual indicator of force applied in a suturing task. They found that, although there was an initial benefit, the assistance deteriorated with time, indicating that the fatigue of checking the visual indicator interfered with the short term effect. A haptic indicator may have overcome this effect as it would not have required the user to continually change visual focus away from the main task. Another area in which it is important not to cause any distraction when supplying ancillary information is in the task of driving a vehicle. A haptic system was developed by ur Rehman et al. in 2005, to provide a vibration warning to the driver when a pedestrian is detected [181]. It is used as an advisory system, in that it can be tuned on the 'safe' side – i.e. warnings can occur when there is just a suspicion, and the user can mediate and adjudicate them.

Another feedback method is the use of vibro-tactile stimulation, as used by Bosman et al. in a 2005 study involving the provision of portable vibration devices to help guide visitors around an unfamiliar building [27]. The concept was that of a gentle tap on the shoulder to indicate a change of direction was required. Also in 2005, Brewster and

---

3 Fitts's Law says that a user moving between 2 targets is linear in time as task difficulty increases (targets get further apart and smaller.)
King used a vibro-tactile interface to indicate progress of some background task, freeing the user’s visual sense for the foreground task [32]. Vibro-tactile displays used in remote interaction applications have the advantage of generally lower cost. They also are usually used in an open loop manner, and as such do not have the stability issues involved in haptic feedback loops [120]. However, they typically do not try to represent the actual physical reality of the remote interaction, but instead, replace it with a vibratory clue to the user that must be learnt and understood. In 2002, Prytherch [176] reported that users given a vibratory stimulus coinciding with an appropriate visual display, will describe a contact between a probe and a surface, rather than a vibration felt in the hand. Force feedback also has the capability to modify a user’s behaviour without cognitive attention. This is not so for vibro-tactile feedback. In 2005, Wagner and Howe investigated these two types of feedback using an experiment designed to try to stop users penetrating a virtual wall with their hand [240]. They found that the force feedback could effectively prevent penetration within 150 ms, a period in which cognitive effects do not come into play. They described vibro-tactile feedback as fundamentally an information source, and can not be interpreted within the first 150 ms and conjectured that this means that vibro-tactile feedback requires an increased mental load over force feedback.

2.5 Interacting with Data

Loftin [136] suggests that mapping complex, multivariate data onto more than one sensory modality might have the capacity to increase human bandwidth for understanding that data. Data representation was one of the earliest applications of haptics. In 1990, Brooks et al. created an application which used haptics to investigate inter-molecular forces and docking behaviour [34]. More recently (2006), Subasi and Basdogan developed improved algorithms for this and found that the haptic technology reduced binding errors [224]. Data can also be associated with haptics in an abstract manner. In 1993, Iwata and Noma [107] mapped scalar values against torque to aid in the interpretation of scientific data. In 1996, Avila and Sobierajski [11] directly mapped the position of a probe to a force computed from intensity values and local gradients at that position. In 2005, Lundin et al. [139] looked at volumetric haptics and ways to create ‘virtual surfaces’ with a data volume. Another method of interacting with volumetric data is to map the gradient of a three dimensionally distributed scalar value to a haptic force [147]. This results in a pattern of haptic ‘grooves’ in space that can be
felt and followed easily by the user. In 2003, Loftin suggested that multi-variate data could be conveyed using a combination of human senses, such as sight, hearing, smell, touch, temperature, taste and vibration [136]. He explains that within these, there are various dimensions, for example an auditory display could use position, amplitude, frequency and timbre. Similarly force feedback can constitute the three spatial dimensions as well as amplitude, vibration and friction.

2.6 Ergonomic Issues

The tight link between visual and haptic feedback can sometimes create the illusion that a real tool is being used on real physical objects. This illusion can be shattered if the user tries to rest a part of their arm or hand on a virtual object and find it is not there. Avila and Sobierajski found that providing places for the user to rest an elbow and wrist can help to maintain this illusion of physical reality [11]. These resting places are also necessary to help reduce fatigue and muscle strain that could be caused by prolonged use of a haptic device. In the temporal bone drilling simulator, described in Chapter 12, we use a padded raised bar for the same purpose. This replaces the patient’s forehead upon which a surgeon will typically rest a wrist during the temporal bone operation.

2.7 Collaboration

The latter part of this thesis concerns the use of haptics in collaborative interactions involving two users sharing a virtual haptic environment over the internet. Te’eni et al. [228] describe collaborative environments as having three commonalities: shared workspace, WYSIWIS (What You See Is What I See), and a private space that can be uploaded to a public space. This has limitations as it is aimed too closely on the more established collaboration style of file and text sharing. The comparison of this work with mine is covered more fully in Chapter 8.

In 1999 Brooks stated that “end-to-end system latency is still the most serious technical shortcoming of today’s VR systems” [33] (page 18). In non-haptic collaborative environments [135] [134] [45], the issue of latency is not so critical due to the loose coupling of human users at each end of the network; actions are interpreted visually by a user before reacting. Haptic collaborative systems, however, can experience serious latency-induced effects and this thesis describes one solution to the problem. In their work in 2000, Basdogan et al. studied the nature of the haptic interactions between people, but avoided the distraction of network latency by connecting two haptic devices.
to the same computer [213]. They created a virtual version of the traditional side-show game of moving a ring along a curving wire. They found that performance increased considerably when haptic feedback was used. Also the subjects reported an enhanced feeling of togetherness when haptics were involved.

The problem of latency-induced instability can be addressed with predictive algorithms. Belghit et al. used 1st, 2nd and 3rd order predictors in their work, reported at EuroHaptics in 2002 [18]. They experienced success in controlling haptic collaboration but found that the algorithms needed to be varied, depending upon the amount of network delay. Gutierrez [88] measured the effect that network quality-of-service issues such as packet loss, jitter and latency have on collaborative haptic interactions and found that they were reasonably resilient to packet loss but sensitive to jitter. Cheong et al. [52] describe a synchronisation scheme that is robust to perturbations in the time delay introduced by a network. Other collaborative haptic environments have been developed [78], [90], [96], [115]. None has achieved the asynchronous long distance collaboration with interconnected virtual elastic components detailed in Chapters 8 - 10 of this thesis. Heldal et al. [95] state that in collaborative virtual environments “…two people cannot hold the same object at the same time. That means that to give to each other an object, one person has to let it go and then the other pick it up.” My work, detailed in Chapters 8, 9 and 10 of this thesis, allows just this type of interaction, and therefore demonstrates that this is not so. Experiments have been done to measure the mechanical impedance of a finger and provide a bandwidth response when stimulated by a mechanical piston [38] [89]. The work described in Chapter 9 of this thesis describes experiments which measure the waveforms created when finger tissue, finger movement, wrist movement and arm movement are selectively added to the system, as well as the cognitive response of the central nervous system. Van Essen and Rovers [236] describe a similar physiological model resembling the network layer model used in the data communications field. They describe a mental layer, physical layer, device layer and network layer.

In 1988, Lawrence [126] investigated impedance-controlled manipulators and found that the introduction of time delays above 30 ms led to system instabilities. In 1993, Colgate et al. [55] studied computer simulated spring deformation. They found that any algorithm for spring deformation that is implemented in discrete time will not result in the force increasing smoothly with deflection. The force will be held at a constant value
until updated in a step-wise pattern. The result of this can be that the simulated spring can generate energy which in turn can contribute to instability.

In 2005, Chardavoine et al. described the Wolverine system, which facilitates distribution over a network [49]. They give a good description of basic concepts of scene-graph structure and mechanisms for rendering and go into detail on the concept of engines which can visit each scene-graph node in turn. The scene-graph design of my work is addressed in Chapter 9.

Linebarger and Kessler [134] provide a good summary of conflict resolution and other concurrency issues in collaborative environments. They state that “Concurrency control can be notoriously difficult to get right, especially in a multi-threaded environment where reads are interleaved with writes” (page 309). The authors divide concurrency methods into pessimistic (disallowing concurrent access to an object), optimistic (assuming concurrent access hardly ever happens, so it can be allowed as long as any inconsistencies are repaired), and predictive (like the optimistic method, but using history to minimise the occurrence of inconsistencies). The work described in Chapter 9 of the thesis is more aligned with the optimistic method, and that in Chapter 13 uses a novel way of using haptics to provide a pessimistic method that is intuitive to the user. It turns out that in use, the haptics combines with natural human politeness to provide a very workable solution. Silverstein and Dech discuss this same effect in [210]. This also relates to the work by Zhang and Furnas [253] who describe users working at different scales on the same objects in a virtual environment. Despite the fact that they do not use haptics, they use terms such as ‘push aside’ to describe techniques available for use. My work on collaborative sculpting (Chapter 13) ‘literalises’ this concept (users can really push their partner’s hand aside) to assist in conflict resolution.

2.8 Medical Simulation

Virtual reality is being increasingly used in the training of medical practitioners [219] [14] [183]. Burdea [38] states that "The present, cadaver-based training cannot possibly allow a resident to acquire all the necessary skills in medical school. A learning curve follows graduation which puts patients at increased risk." Laparoscopic procedures are particularly difficult to learn as the motor skills required are not intuitive [229]. Van Dam et al. state that “evidence indicates that humans have one visual system for pattern recognition and a separate one for the hand-eye coordination involved in motor skills”
Assessing a student’s skill in surgical training is traditionally performed via subjective means. Virtual reality provides the potential to introduce an objective measure of performance. In 1995 Ota, Loftin et al. produced a laparoscopic virtual reality surgical training system that used fuzzy logic to determine the skill levels of trainees [168]. In 2002, Dev et al. developed a surgery workbench that included haptic feedback. They state that “Anatomy and surgical procedures require both a conceptual understanding of three dimensional anatomy and a hands-on manipulation of tools and tissue” [66] (page 446). In 2005, Srimathveeravalli and Kesavadas [218] showed that visually displaying the difference between a student’s applied force and that of an expert could improve learning in a motor skill. In 2006, Strom et al. [223] found that haptic feedback is important in the early training phase of skill acquisition in image-guided surgical simulator training.

Active research is being carried out into the simulation of physical tissue deformation and cutting. Montgomery et al. developed the Spring [150] system and incorporated it into a surgical simulation environment for teaching hysteroscopy [151]. Song and Reddy [216] describe surface mesh cutting. When a surface node is deleted it is replaced by two nodes. A spring model is added to retract the newly created nodes. The tissue cutting method described in Chapter 10 of this thesis uses a simplified version of this technique.

A training system for temporal bone drilling is addressed in Chapter 12. Other work in this area [68] [94] has used similar techniques, but concentrates on the physical interaction of the drilling burr with the bone. The work covered by this thesis is aimed at producing a broader training application for the procedure. The use of simulators for training is discussed by [186], who explain that one of the main benefits of using a virtual reality system for training is that it allows the student to explore and learn from their mistakes. This exibility allows students to gain exposure to a wide range of situations, and gain an understanding of the rationale for each step in a procedure. Another medical use of haptics and simulation is in the area of surgical planning [231]. By using patient-specific medical scans to create 3D models, it is possible for surgeons to discuss and explore options, using a touchable virtual model of the patient before surgery is commenced. An essential component of both surgical training systems and surgical planning systems is the 3D anatomical model itself. This is usually created from medical scans, such as MRI or CT, and is referred to as segmentation. Work is
progressing on the automatic and semi-automatic generation of 3D models from medical scans. Although this is outside the scope of this thesis, it is worth noting that some interesting haptic research [238] [239] has been performed to help a user to perform the segmentation with the aid of haptics, allowing them to ‘seed’ the internal areas of identified body organs, and feel the boundaries that are generated as a result.

2.9 Deformation

Sculpting of deformable (i.e. pliable) objects is covered in Chapter 3 and Chapter 13. Other work in this area includes [38] who describe a method of plastic deformation which involves an initial small elastic deformation zone. Forces grow linearly with deformation until this initial zone is passed, when the deformation becomes permanent. In 2006, Ohguchi [165] reported that plastic deformation can be either temporally dependent or temporally independent. My investigations into both of these styles of plastic deformation are described further in Chapter 3 of this thesis. Other work [59] has incorporated two HapticMaster [235] devices to provide a two-handed system to allow designers to make models with a sculpting and scraping action. In 1991, Galyean and Hughes [73] presented a voxel model, overlaid with a surface generated from a ‘marching cubes’ algorithm. In 2005, Prior and Haines [175] described the use of a point-shell based tool for voxel erosion and deformation. A proximity agent is used to cull out voxel objects which are not close to the point shell. Snibbe et al. [215] have investigated non-physically based dynamic models for sculpting and found that users could sculpt deformable objects more smoothly if a synthetic damped spring was attached to their finger position. Other application areas within the arts have made use of haptic user interaction. For example, in 2002, Lin et al. [133] described a haptic painting system with different shaped brushes.

---

4 Voxel model; a 3D model comprised of small cubic elements, named to parallel the word ‘pixel’ for rectangular elements in a 2D model.
Chapter 3.

Simulating Reality

3.1 Introduction

This paper, Virtual Artisan’s Workbench was presented at the Simulation Technology and Training Conference in Sydney, 2000.

Contribution: I was sole author on this paper. I investigated several different techniques of allowing deformation\(^5\) of a surface and developed and implemented the algorithms used in the interaction of the virtual tool with both deformable clay and deformable lines. I also developed and implemented the user interface and surface texture painting algorithms. I performed further work outside of the scope of the publication, and this is detailed in the discussion section of this chapter (section 3.3). The concept of passing a subset of vertices from the graphics thread to the haptics thread drew on unpublished earlier work by a colleague, John McLaughlin.

Plastic deformation of an object is a technique that had not been addressed in the haptic toolkits available at the time of writing. This paper describes an investigation into a method for performing plastic deformation in a haptically enabled virtual environment. It goes on to describe an application for artists that was built around this feature. This paper lays the foundation for a collaborative version of the application, a topic which is addressed in Chapter 13 of this thesis.

\(^5\) Deformation - change in the shape or dimensions of a body, resulting from stress
It should be noted that there have been advances in haptic sculpting of virtual clay since the publication of this paper. Of particular note is the approach of using a voxelated volumetric model to handle the haptic properties and then covering this with a triangulated surface to provide the visual appearance [202].

The conference’s audience comprised representatives of government departments, the military and a small number of medical administrators and practitioners, all with an interest in using virtual reality in training systems.

Note that references to “Magma” concern a programming API that is now marketed by Reachin AB of Sweden [180] under the name of Reachin API™.
3.2 Publication 1: Virtual Artisan’s Workbench

Chris Gunn
CSIRO
Building 108, North Road, Australian National University, ACTON, ACT 2601.

Keywords: deformable, haptic, sculpt, virtual environment, 3D

Abstract

This paper describes an experimental system for simulating the operations of sculpting and surface painting. The system is called a "Virtual Artisan's Workbench" (VAW). The VAW is implemented using a desktop virtual environment, called the Haptic Workbench (HWB), developed by the ACSys Cooperative Research Centre. The HWB provides co-located 3D graphics and haptics; that is, with appropriate programming, a user can both 'see and feel' a 3D model. The objective of the VAW was to develop a system that enabled 'virtual' clay modelling which has applications in areas as diverse as manufacturing, architecture and animation. This paper describes the hardware and software structure of the VAW, as well as limitations found in simulating deformable surfaces, and investigations into haptically based user interface controls.

3.2.1 Introduction

Immersive virtual environments provide a user with the ability to interact with a model or scene in a much more realistic way than was possible with conventional flat-screen displays and 2D input devices such as a computer mouse. The ability to render a scene in 3D stereo vision, along with head-tracking and haptic (force) feedback, allows the representation of virtual objects with a realism that permits the user to temporarily “suspend disbelief” and operate on the objects as if they were real world objects.

For the last three years, the Cooperative Research Centre for Advanced Computational Systems (ACSys) has been experimenting with the application of immersive environments and haptic feedback in its Virtual Environments Laboratory in Canberra. Haptic feedback, in this context, refers to the transfer of force from a computing system to the user’s input device. The device currently used in our laboratory is the SensAble Technologies [201] PHANTom robotic arm. This is used as a 3D pointing device for input of positional information to an application, but can also receive forces from its motor driven cables. These forces can be configured to simulate the ‘touching’ of virtual
objects. When combined with a graphical representation of the objects being touched, the simulation can be quite convincing. The force feedback can also be used to simulate textures, springs, gravity, magnetism and vibrations.

The Virtual Environments group’s research has so far been centred on a configuration called the Haptic Workbench, described in detail in [221]. The arrangement of monitor, mirror and polarised stereo shutter glasses creates a small-scale 3D virtual environment into which users can directly place their hands, to manipulate real-world, physical devices that have virtual representations as tools in the virtual scene.

The co-location of the graphics and haptics has been an important achievement in presenting a believable simulation [25]. The robot arm is located so that the tip of the arm (the tool) is co-located with the graphic representation of a tool in the 3D scene. To avoid real-world objects (the users arm and the PHANToM) obscuring the scene, we view the screen reflected from a mirror. The user manipulates real-world objects behind the mirror, but sees virtual objects moving in the mirror. The CSIRO Haptic Workbench has the ergonomic advantage that users are seated and working at a normal desktop height, in the normal posture that a technician might take up at a real workbench.

The mirror is not obtrusive and the shutter glasses are quite comfortable. This capability also provides considerable challenges for the application developer, especially in the area of haptic feedback. This is because the refresh rate required to provide a convincing ‘feel’ to objects is such that there is a very small amount of time available to
perform the necessary force calculations. For this reason, physical accuracy must sometimes be sacrificed to permit faster approximations.

3.2.2 Virtual Artisan’s Workbench
To experiment with deformable surfaces and to also provide a novel demonstration of the capabilities of the equipment, we developed an application that simulates the sculpting of a ball of clay. This also provided a useful test bed for a Deformable Shapes library of C++ classes, which we are developing, specifically for use with the Haptic Workbench. The challenge in this project was to provide a realistic feel to the clay as it was being pushed or pulled. The force feedback that is provided by the surface-touching algorithms developed for rigid surfaces would not be suitable for surfaces that deformed when they were touched. This is because as soon as the tool penetrates the surface, it retreats and the penetration (and force) drops to zero. The application was extended to incorporate drawing on and painting the virtual object.

Other work has been done, see [202], providing a sculpting environment using a voxel-based method. This work differs in that it is surface based.

3.2.3 Approximations Required for Haptic Feedback
All objects react to a touching force. The energy of the user input is usually absorbed by the whole object moving and/or being deformed. When a person touches a rigid, real world object a resistive force equal and opposite to theirs is felt. The returned force onto their hand changes from zero just before they touch the wall, to a finite force as soon as they touch it. If such a step function was implemented in a computing system, with unavoidable calculation delays, there would be a risk of causing instability. To avoid the vibrations that this delay would introduce into the system, we must approximate the step function with something less accurate. This could be one that allows the user to penetrate the wall sufficiently to allow the force to ramp up over a distance, but not so much that it destroys the illusion that the object is solid and rigid. To enhance this illusion, we can keep the graphic representation of the user’s tool exactly at the surface, with no penetration, while allowing the actual haptic tool to depart from the graphic representation momentarily as described in [254]. This is the method used for surface haptic rendering used by the Magma Graphic/Haptic Library, co-developed by ACSys [2] and Reachin Technologies [180]. This library is used as a basis for the Deformable Objects Library.
When modelling objects that are deformable, we must make similar approximations. As a real tool contacts a real deformable solid (say, a ball of potter’s clay), it comes into contact with millions of clay molecules. It pushes these molecules in the direction of movement until their combined inter-molecular attraction balances the force applied to the tool. When modelling this situation in a computer we are confronted by two major problems. The first is the limited time we have to do our calculations. This is worse than the situation with the rigid object described above, since the calculation involves re-positioning the surface of the object as well as calculating the returned force on the tool. The second problem is that we cannot afford to calculate the same number of points as molecules in the clay. The time taken to search the millions of points to see which are colliding with the tool would be prohibitive. We must therefore make a gross approximation, by using a sub-sample of points to represent the object. The number (and therefore, density) of points that can be accommodated depends on the processing speed of the computer. In our laboratory, we have an SGI dual processor Octane that starts to degrade at about 5000 points.

### 3.2.4 Prediction

Another method that can be used to improve the performance of a haptic feedback application involves the use of a predictive algorithm to search for candidate points that may be deformed. This is possible because many haptic programs use a separate thread for the haptic and graphics rendering. The graphics rendering need only refresh at 30 Hz, while the haptics thread must keep close to 1000 Hz. The track of the user’s tool is predicted during the slower graphics thread. The points that it could possibly reach in the next graphics cycle are collected and passed to the haptics thread. The haptics thread then has a much smaller list of points to search to detect collisions.

### 3.2.5 Types of Deformation

Surfaces can be deformed plastically or elastically, with typical deformations having a combination of both. With 100% plastic deformation, the surface recedes as the tool is pushed into it and does not rebound when the tool pressure is removed. With 100% elastic deformation, the surface similarly recedes with tool movement, but returns to its original shape as the tool is removed. This return is controlled by the tool’s position as it is withdrawn, always remaining in contact with the clay.

A surface exhibiting a proportion of both effects will recede with the tool’s forward movement, but lag behind the tool as it is withdrawn.
As the user pushes the tool forward, they would expect to feel some resistance. There are at least two ways to model this resistance. The first is to calculate an opposing force proportional to the velocity of the tool. This is in effect a damping force. As long as the tool is moving forward, the user feels a resistance. As soon as movement stops, there is no force. In a situation involving elasticity, a force should also be felt when the tool is stationary and in fact moving backwards. The calculation of this force would involve some complex modelling of the material properties of the surface.

The second method of calculation does not use the tool’s velocity, but instead uses the tool’s position relative to the vertices of the model to calculate the force. It necessitates the approximation of allowing the tool to ‘sink below’ the surface somewhat; i.e. the tool edge penetrates the surface. In most situations, this is not discernible to the eye and is therefore an acceptable approximation. The advantage of this method is that since the depth of penetration can be used to calculate the force, a force can be applied no matter what direction the tool is travelling. The user feels some resistance when they are pushing forward, stationary or moving back. The amount of resistive force can be proportional to an elasticity factor. Similarly the amount of penetration relative to the amount the vertices are moved can be varied. This also enhances the illusion of elasticity. As the tool is withdrawn, the penetration reduces to zero along with the force. Although the graphically rendered vertices do not ‘spring back’, an element of elasticity is perceived haptically. This is only effective for surfaces where plasticity is dominant over elasticity.

A truly elastic surface can be simulated with the CSIRO Haptic Workbench using an algorithm that is more computationally expensive. It involves calculating the tool’s movement during each graphic cycle and checking for intersections with triangles on a triangulated surface. When an intersection is detected, the tool’s movement is tracked each haptics cycle, and forces are calculated depending on the instantaneous distance of the tool tip from the original contact position. Because of the necessity to calculate triangle intersections, there is a greater processing load, so that, at current processor speeds, calculations can only be performed for a single point tool (sharp tip). The single point contacts one point on the model and the deformation occurs on surrounding surface with a concentric, decreasing influence. However, the elastic behaviour is much more convincing, allowing the surface to depress and return to its original shape with apparent physical correctness.
3.2.6 Magma (now called Reachin API)

The Deformable Shapes Library is built over the Magma Haptic/Graphic Library. The Magma library has a C++ and VRML interface, the former allowing complex haptic operations to be implemented, while the latter is easy to use for straight-forward scene building. Magma’s event model involves a field network, and classes (e.g. Cube, Sphere) have their properties (e.g. radius) stored in fields. These fields can be interconnected via routes. One field can update another and so on along a chain of fields. Much programming in C++ with Magma involves inheriting from a Magma class and adding new fields. Much programming using the VRML interface of Magma involves creating and positioning objects and routing them to each other.

3.2.7 The Deformable Shapes Library

The Deformable Shapes Library is an object-oriented library of C++ classes, some of which inherit from Magma classes. It contains several deformable shape classes, some of which have their common functionality abstracted out into common base classes. It also contains several supporting classes. The Deformable Shapes classes can be created and manipulated from C++ or VRML. First time users should begin with the VRML interface, as it is much simpler. The approximations described earlier are used for the plastic/elastic shapes that inherit from the abstract base classes. A 100% elastic model is provided by the Membrane class, using the single point tool mechanism.

Deformable Surface

A DeformableTriSurface is a triangulated surface that exhibits plastic deformation when a spherical tool presses it. It has the ability to be deformed from both sides. This produces the effect that a surface can be stretched from the inside as well as pushed from the outside. It has fields that can modify its behaviour in several ways:

It can be set to respond to sculpting by a spherical tool, or alternatively it can have individual vertices moved by a ‘picking tool’ (when in PICK mode the vertex about to be picked can be highlighted to assist the user).

It has fields of vertex mass, stiffness and cohesion. These effect the reaction of the surface when contacted by the sculpting tool. The algorithm, detailed in [255], combines these in a spring/mass model.

It has a symmetrical editing mode, which will reflect any changes across a configurable plane.
It also has the ability to restrict the deformation to a certain direction. This can be useful if you want deformations to occur perpendicular to a plane, for example.

**Deformable Line**

A DeformableLineSet holds a multi-segmented line that can be modified by sculpting or picking. In PICK mode, the nearest vertex can be highlighted by the application before being picked. In sculpt mode a resistance is felt as the sculpting tool presses against the line. It feels somewhat like bending a piece of soldering wire. It also works on a spring/mass model, and has fields of stiffness, vertex-mass and cohesion that can change the behaviour under deformation. The algorithm is a simplified version of that used for the DeformableTriSurface.

**Membrane**

A Membrane is a deformable IndexedFaceSet that is elastic. It has some similarities to the DeformableTriSurface class, but its underlying implementation is completely different. It also has some important differences in its behaviour. The most obvious difference is that it is elastic, springing back to its original shape when the deforming tool has been removed. A more subtle difference is that it reacts to a single point of contact with the tool, not a tool sphere, as does the DeformableTriSurface. This means that using this algorithm we cannot have tool tips with a radius, or a shaped tool tip. The deformation of the surface radiates outwards from the point of contact in an inverse squared relationship to distance.

The Membrane class implements the elastic deformation algorithm, mentioned earlier. It has fields which modify its overall elasticity, and the elastic resistance in the normal direction and tangential direction to the surface.

The deformable shapes library also contains several support classes, servicing the algorithms contained in the deformable classes.

**3.2.8 The Virtual Artisan’s Workbench Application**

The Virtual Artisan’s Workbench application was designed to demonstrate some of the features of the Deformable Shapes Library. It presents the user with a ball of virtual clay and a row of haptic buttons, a ‘compass’ and a tool sharpener. (See user interface issues, below). Several of the buttons are arranged in a radio button panel, allowing the choice of one of several modes.
In sculpt mode, the virtual tool becomes a rod with a spherical tip. The tip radius can be adjusted by inserting the tool into the virtual ‘sharpener’ and twisting. As the tool contacts the clay, a subtle force is felt and the clay is pressed inwards. The tool can be inserted inside the ball of clay by quickly stabbing the surface. Once inside, the surface can be gently worked outwards. In this way, clay can be dragged out as well as pushed in – an advance over real clay!

If the user selects the symmetrical editing mode, any sculpting they do on the left of the model is mirrored in the right, and vice-versa. This is particularly useful for sculpting faces.

Another of the radio buttons selects a paint mode. A series of virtual paint pots appears alongside the clay. The tool changes to a paintbrush. The user dips the brush into a pot to select a colour and runs the brush along the surface to paint it. In this mode, the surface is hard (fired clay?) and will not deform. Once painted, the sculpt mode can be re-selected and the paint will deform correctly along with the surface – something else that can’t be done with real clay! This ability is a consequence of the paint being implemented as a texture map on the surface.

Another mode allows lines to be drawn anywhere in the 3D workspace. The user presses a real button on the PHANToM tool to draw, much as in a 2D mouse drawing system. A second drawing mode is more interesting. When this is selected, the virtual tool becomes a ballpoint pen and behaves like one. I.e. when it touches something it draws a line on it, when it is lifted off, it stops drawing.
In “pick” mode, the tool becomes a sharp pointer and as it is moved, the nearest vertex, whether on the clay or on any drawn line, is highlighted. When the user presses the tool’s mouse button, the highlighted vertex is moved with the tool’s movement. Using this mode it is possible to do fine editing on individual vertices.

There is also a smoothing mode that will attempt to modify triangles to become equilateral. This mode also uses the tool metaphor, working only on the triangles that the tool is brushed across.

The output of the user’s artistic endeavours can be saved in VRML format, and therefore is available to be imported into a variety of other modelling packages, as well as viewable in most web browsers.

3.2.9 Limitations

Sculpt is currently being installed in the CSIRO’s Discovery Centre, in Canberra. This does not mean that it is a completed project, however, and there are still some limitations in its current state.

The requirement to complete all necessary haptic calculations within about 1 millisecond presents a severe restriction on the type of algorithm that can be used to calculate the forces. An approximation to real material behaviour can only be achieved at present. The prediction of likely interacting vertices during the (slower) graphic thread alleviates the pressure on the haptics thread somewhat, but in turn presents an efficiency problem of its own. To predict likely vertices, it must search all vertices. If this takes too long, the tool rendering is slowed down such that movements appear jerky. At present processor speeds, sculpt can handle about 5000 vertices. A solution to this is currently being investigated (see Future Work).

As a surface is sculpted, the triangles are stretched. This can typically result in some long, thin triangles, with vertices a long way apart. Since the tool reacts to vertices, it can sometimes pass right through the centre of such a triangle without contacting any vertices. Also, as vertices get stretched further apart, less detail can be sculpted into that area of the model. For example, when sculpting a face, the area of mouth and lips need too much detail for the system at present, to provide.

Since the force feedback depends on vertex collisions, the force felt by the user is sensitive to vertex density. The deformable surface’s parameters must be carefully tuned.
to the vertex density of the model. Bad tuning can produce effects which vary from the tool passing right through the surface, to unpleasant vibrations as the tool touches it.

### 3.2.10 User Interface Issues

User interfaces in 3D immersive environments are problematic. The user typically needs to change modes or select a value while they are involved in manipulating their 3D objects. In a 2D, windowing environment, they position the cursor ‘over’ an on-screen button and click the mouse button. In a 3D environment, the ‘cursor’ is a tool tip. There is usually no plane onto which to put buttons and sliders, and therefore no concept of putting a cursor ‘over’ a button on a plane. However, there are several different ways of providing user interface interaction in 3D, all of which have advantages and disadvantages. Some of those considered for the sculpt application were:

- a) Providing a 2D panel holding traditional sliders and button. The tool becomes a 2D cursor as it enters the XY extent of this panel. We thought that this change of modes would be confusing to the user.

- b) Using speech recognition and therefore doing away altogether with the need for GUI widgets. We couldn’t find a suitable, affordable, UNIX based system for this.

- c) Creating fully 3D widgets embedded in the 3D scene at convenient points. This is the UI that we implemented in sculpt.

The 3D buttons were given haptic properties and allowed to move slightly when prodded with the tool.

However, since this movement required quite complex modelling of the dynamics to avoid feedback problems, it was considered overkill to provide this processing for each and every button. An attempt was made to simulate the depression of buttons using the traditional illusion used for ‘3D’ buttons on 2D hardware; i.e. that of changing a top-left/bottom-right highlight strip around the button to make it appear depressed. However, we discovered that the stereo vision provided by the shutter glasses defeated the illusion, and the buttons just did not appear to be depressed. Finally, as a compromise, we came up with a background panel for the row of buttons. When the user pushed a button, the section of panel surrounding the button would pop forward slightly, thus giving the impression that the button had moved backwards. Because the object being touched is not the object moving, there are no feedback problems.
To allow for colour choice in the painting and drawing modes, we needed to provide some sort of colour pallet that was compatible with the 3D environment. Since we had the ability to provide ‘real’, touchable objects, we followed the theory that “real is best” and created a set of 3D paint pots. They had hard, glass, exteriors and soft, gooey paint inside (with a bit dribbling over the edge). The idea was that the user dips the paintbrush into the jar of paint to change colours. What could be simpler? However, we didn’t count on the users being computer-literate. They knew that they were working on a computer system, and the jars were not really there, so they would often try to pass through the side of the jar instead of dipping into the top. Of course, they would just bump into the glass. In hindsight, it is much more convenient to enter a jar from any direction, and if real paint pots could be built that way, it would certainly be an improvement! So the lesson we learnt, was “real is not always best”!

We have had similar problems with the 3D buttons. Because the PHANToM tool has a small clicker switch near its tip, we found that users would position the tool near the 3D button and try to click the clicker, instead of just pushing the button in the 3D scene. It was hard for them to change from the ‘WIMP’ metaphor, into one where you push buttons as you do in the real world. We found that even when we told a first-time user to “press the white button” they would interpret this to mean position the virtual tool near the widget and press the clicker. Discussions are continuing in our laboratory on whether to give up fighting against ingrained 2D thinking and provide a more 2D-like interface for buttons and sliders.

3.2.11 Future Work

Work is currently progressing on introducing a spatial indexing mechanism into the vertex search algorithm. This will trade off the extra cost of calculating a vertex’s spatial index when it is moved, against the benefit of having a quicker search time to predict likely vertices for collision during the next graphics cycle. Since the spatial index algorithm will be parameterised, the optimum trade-off between the two should be found by varying the spatial index cell size.

A possible solution to the problem of overly stretched triangles would be to re-triangulate (sub-divide) any that exceed a certain size. To do this on the fly, however, would interrupt the normal sculpting processing to the extent that it would probably be felt as jerks or vibrations. A compromise may be to give the user via a ‘re-triangulate’ button, so that sculpting is paused while the operation proceeds. A better solution, and
one more in keeping with the tool metaphor, may be to have a ‘re-triangulate’ tool mode. In this mode, the tool is brushed on the surface and ‘paints’ on extra triangles. Then a user could decide on an area to paint on extra fine triangles, for detailed work.

==========================================
3.3 Discussion

In his dissertation, Hinkley wrote

Virtual manipulation poses a difficult dilemma: one wants virtual objects to violate reality so that one can do things that are not possible to do in the real world, yet one also wants virtual objects to adhere to reality so that the human operator can understand what to do and how to do it. The interface design challenge is to find ways that real and virtual objects and behaviours can be mixed to produce something better than either alone can achieve [212]; and part of this challenge is to discover interaction techniques that do not necessarily behave like the real world, yet nonetheless seem natural. This leads to a key point: to design interaction techniques which meet these criteria, short of taking wild guesses in the dark, the interface designer needs to understand the human.

[97], Chapter 1, page 21.

This paper describes one such combination of the simulation of reality (such as the provision of touchable surfaces on objects) with convenient violations of that reality (such as the ability to pass through surfaces when required). The model has plastic deformation, but only deforms locally to the interaction, where the sculptor can see it. Without conservation of volume, unexpected bulges are not produced in areas of the model that are not being touched.

3.3.1 Elastic Deformation

Before discussing the plastic deformation technique in more detail, it is helpful to understand a common method of supplying elastic deformation to a simulated surface. Pure elastic deformation is said to occur when a surface will return to its original, pre-deformed shape when the deforming external load is removed. The Reachin API™ toolkit [180] provides an approximation to this capability through its Membrane component. In this case, an original, undeformed surface is maintained in computer memory, while a deformed surface is rendered graphically on the screen. The force returned to the user via the haptic tool is based on the difference between the original touch point on the undeformed surface and the current haptic device position beneath the surface. As the haptic tool is pushed deeper below the original surface, the difference between these positions increases and this causes the magnitude of the force to become proportionally greater. This difference in the position is also used to reposition the vertex nearest the haptic tool tip. The surrounding vertices are also
repositioned, but with a delta that is inversely proportional to the square of the distance from the haptic tool tip. Since the polygons comprising the visible surface are built upon these vertices, the surface itself is redrawn accordingly. This causes a depression around the touch point and results in behaviour resembling a rubber sheet or membrane touched with a fine point. Parameter adjustments can vary the extent of the deformation and also the stiffness of the surface (i.e. the force returned as a proportion of the penetration of the tool). This system, while producing a satisfactory result for purely elastic deformations, is not suitable for the simulation of surfaces with any degree of plasticity as well as elasticity.

3.3.2 Plastic Deformation

In purely plastic surfaces, any deformation remains permanently after removal of the deforming force. Many real-world materials have both plastic and elastic properties, springing back partially to their original shapes, but maintaining some degree of permanent (or at least long-lasting) deformation. Such plastic deformation can be time dependent or time independent, or a mixture of both. When plastic deformation is time-dependent, the surface’s permanently deformed shape progressively approaches its elastically deformed shape over time, as long as the deforming force is maintained. On the molecular level, elastic deformation corresponds to the molecular bonds being put under stress. Time-dependent plastic deformation occurs when these bonds progressively break and reform over time. When plastic deformation is time independent, the molecular bonds break at the instant that they are stressed, during the initial deformation, and the surface attains its permanent deformation immediately. In both cases, there may be bonds that are stressed but have not broken, causing some elastic rebound in the material when the external load has been removed, resulting in some degree of elasticity. This is discussed in more detail in [165].

The experimentation for the simulation of virtual clay used in the sculpting application described by this paper, involved a comparison of both methods of plastic deformation. In each case, the surface of the clay is represented as a mesh of interconnected triangles, defined spatially by their vertices and topologically by a list of triangle interconnections. The haptic tool tip is represented both graphically and mathematically as a sphere of finite radius.
3.3.3 Simulation of Time Independent Plastic Deformation

The simulation of time independent plastic deformation involves tracking the tool tip motion and comparing it with a list of triangle vertices. Two processor threads are involved in the deformation simulation, the graphics thread and the haptics thread. The graphics thread cycles at approximately 30 Hz and handles graphic rendering plus program logic. The haptics thread must run at 1000 Hz to avoid vibrations in the haptic tool. It handles force calculations and any program logic that directly changes haptic force. Because the surface is being deformed by user interaction through the haptic tool (i.e. virtual touch of the surface), tool motion directly and immediately affects the force feedback, and as such must be calculated during the haptics thread. However, due to the stringent requirements of the haptics loop, it is necessary to optimise both this calculation and the search for colliding vertices. To accomplish this, the graphics thread does a preliminary search for candidate vertices in the close vicinity of the haptic tool, and passes the list to the haptics thread for use during its next 33 cycles.

When the tool tip’s motion results in a vertex from the candidate list penetrating the tool tip (actually a sphere), it is repositioned via an algorithm within the haptics thread. The new position is used for further collision and deformation calculations, but is not rendered graphically until the next graphics cycle encounters it. The repositioning vector is calculated to be in the direction of the nearest tool tip surface, radially from the tool tip centre, but is scaled to be only a proportion of the distance towards that point. This is an approximation at physical reality, but it has the advantage of providing a convenient means of calculating a haptic force representing the surface stiffness. The movement of the vertex is given by the following equations:

\[
D = R - |\vec{V} - \vec{P}|
\]

\[
\vec{M} = \frac{(\vec{V} - \vec{P}) \times D \times S}{|\vec{V} - \vec{P}|} \quad D > 0
\]

\[
M = 0 \quad D \leq 0
\]

where \(D\) = penetration distance
\(\vec{M}\) = movement of the vertex
\(\vec{V}\) = original vertex position
\(\vec{P}\) = tool tip centre position
\(S\) = surface stiffness factor; 0.0 < \(S\) < 1.0
\(R\) = tool tip radius; \(R\) > 0.0

(3.1)
It can be seen from equation 3.1 that the movement is scaled by $S$, the surface stiffness factor, which is always less than one. Since $D \times S$ must therefore be less than $D$, the incremental movement of the vertex, $M$, will always be less than the penetration distance of the vertex inside the tool tip, thus ensuring that the vertex is never pushed completely outside the tip. The consequence of this is that, given no user movement of the tool tip, there is always some residual penetration of the vertex inside the tool tip at the end of each update cycle. (Note that in equation 3.1, the penetration distance is the penetration of a surface vertex into the spherical tool tip, not the penetration of the tool tip below the clay surface.) These penetrations are used to calculate the force returned to the haptic device, according to equation 3.2. This has parallels in the penalty based approach described by Zilles and Salisbury for calculation of forces where a single point virtual tool penetrates a simulated stiff surface [254]. The difference here is that we have a finite sphere as the haptic tool and we are using the penetration of each of a number of surface vertices into the tool instead of a single point tool into the surface.

\[
\vec{F} = \vec{M} \times \frac{(1 + R)}{(1 + R - D)}
\]

where \(\vec{F}\) = force  
\(\vec{M}\) = vertex movement  
\(R\) = tool tip radius  
\(D\) = penetration distance

(3.2)

The use of a sphere of finite radius allows for a much wider range of sculpting actions than would be possible with a single point. Chen and Sun [51] investigated a similar approach for surface interaction in body-based collisions, but without the plastic deformation of the model. The system we use also allows for some surface cohesion; vertices which are neighbours of the moved vertices are calculated to move to some degree as a result of the sculpting operation. The degree of cohesion is an attribute of the surface as a whole. To allow this calculation, each vertex contains a list of its immediate neighbours. When a vertex is repositioned, its neighbours are visited in turn and repositioned themselves according to connective distance away from all directly manipulated vertices.
3.3.4 Time Dependent Plastic Deformation

The simulation of time dependent plastic deformation was built as an extension to the Reachin API elastic deformation class, Membrane. As mentioned in section 3.3.1, the Membrane class holds the original, permanent vertex positions in memory while the temporary, deformed vertices, and resulting surface, is rendered to the screen. The time-dependent plastic deformation algorithm entails progressively moving the permanent vertex positions towards the temporary, elastically-deformed positions, over time, as long as the deforming force is maintained. When the deformation force is removed or relaxed, the rendered surface returns towards these new positions, not the original ones, resulting in some permanent deformation. The user’s original surface contact point is also detected and moved incrementally towards the user’s current contact point on the deformed surface over a period of time. Since the force returned by the haptic device is dependent on the difference between these two points, this force also reduces with time, as the distance between the two reduces. The result is that after a user deforms a shape a certain distance, the shape will partly return to its original form after release of the user’s contact. Consequently, as the user deforms the shape, they feel a resistive force, which progressively relaxes over time. The degree to which the permanently deformed shape approaches the temporary, elastically deformed one is proportional to elapsed time while under elastic deformation. It is also proportional to a plasticity factor, which governs the incremental step in the movement of the permanent vertices towards the temporary ones.

Note that the haptic tool is touching the elastic surface, not the permanent one. This elastic surface is not changing with the progression of the permanent surface towards it. It is therefore possible to avoid calculating each new plastic surface within the faster, haptics cycle. Experimentation found that, although the force returned by the haptic tool was dependent on the difference between the two surfaces, modifying the one that is not being touched during the slower graphics cycle still resulted in a smooth, vibration-free result. The reason for this is that there is no momentary loss of contact with the surface as it moves and the 30 Hz step-wise changes in haptic force are sufficiently small to be outweighed by other damping factors in the system. The permanent movement of the vertex position during each graphics cycle is given by equation 3.3.
\[ \vec{V}'_i = \vec{V}_i + \left( \overrightarrow{DV}_i - \vec{V}_i \right) \times P \]
where \( \vec{V}'_i = \) the new \( i^{th} \) permanent vertex
\( \vec{V}_i = \) the \( i^{th} \) permanent vertex
\( \overrightarrow{DV}_i = \) the \( i^{th} \) elastic deformed vertex position
\( P = \) a plasticity factor; \( 0 < P < 1.0 \)

3.3.5 Deformation Used in the Sculpting Application

The sculpting application was implemented using the time independent algorithm for surface deformation. It was considered to more closely characterise the behaviour of real clay, which deforms immediately to any manipulation. As well as allowing deformation of the clay surface, the same vertex manipulation algorithm was applied to 3D lines that could be drawn in the workspace. The 3D lines are a series of line segments joining a number of vertices and, as such, can be treated in the same way as the list of vertices making up the surface.

3.3.6 Surface Texture Painting Algorithm

The algorithm used for painting the clay surface involves mapping an image texture to the surface using pre-defined 2D texture coordinates and mapping points in the image \((s,t)\) to 3D vertices on the model \((x,y,z)\) (figure 3.3).

Figure 3.3: Mapping from texture coordinates to 3D vertices
The rendering algorithm then uses the texture coordinates to draw the image on the surface, regardless of its deformed state. As the haptic tool tip touches the surface, the texture coordinates of that surface-contact point are passed to an algorithm which changes the colour of the pixels at the corresponding point on the image. It also changes the colour of a number of surrounding pixels, depending on the selected size of the paint brush. Whenever the paint brush size is adjusted by the user, an offset array is pre-calculated and stored. This optimizes the selection of the image pixels to be changed during the painting action.

Care must be taken to provide a fairly uniform mapping of texture to triangles on the surface. Also, with a solid shape, the seam, where the edges of the texture meet, must be treated as a special case. In the sculpting application, the seam runs vertically in the y-z plane. Texture coordinates immediately on one side of the seam have a value close to (0.0, t) and on the other side of the seam have a value close to (1.0, t). The triangle tessellation algorithm must ensure that no triangles span the seam, as this would result in a single triangle mapping across nearly the whole width of the image, from 0.0 to 1.0. The algorithm seeks out such offending triangles and subdivides them to align their edges exactly on the seam. However, this then results in seam-based vertices being associated with triangles to their left having texture coordinates close to (0.0, t) and other triangles to their right having texture coordinates close to (1.0, t). It is necessary, therefore, to allow each vertex to have two associated texture coordinates, labelled for convenience, “left’ and “right”. The rendering algorithm knows which triangle it is dealing with and can determine which texture coordinate to retrieve from the vertex to perform the correct texture mapping.

Painting can therefore occur onto whatever image is currently mapped to the surface. By default, the sculpting application starts with an image consisting of a single colour, approximating the colour of clay. However, it is also possible to map any other image to the surface. Since this image is repeatedly rendered during each graphics cycle, it can be repeatedly changed to produce animation. I produced a variation on the sculpting application applying a video texture to the sculpting surface, with the video retrieved from a stored file. At each graphics refresh cycle, a new video frame is retrieved from file and applied as an image to the surface. Using this technique, it is possible to sculpt a clay model rendered with a video movie.
3.3.7 Symmetrical Editing

The symmetrical editing feature was added to the application when I noticed that users would often try to sculpt a human face. In 1865, Charles Darwin noted in his book, The Expression of Emotions in Man and Animals [60], that humans seem to have an innate ability to recognize other human faces. He argued that facial gestures express critical social cues and the ability to understand these may have evolved over time. It may be for this reason that subtle irregularities in the artificial representation of human faces appear to be much more noticeable than irregularities of a similar magnitude in other shapes. It may be because of this that users of the sculpting system were often dissatisfied with the facial sculpture that they produced. Freeform sculpting of symmetrical objects was observed to be quite difficult, and faces are particularly problematical examples of this.

The symmetry feature was added to assist users to sculpt objects which are symmetrical along the x axis (i.e. on either side of the y-z plane), such as human faces. As the sculpting action causes each vertex to move, its ‘twin’ vertex is selected from a pre-computed mapping across the plane of symmetry, and the movement is reflected across the plane onto this vertex. The mapping is bi-directional, so that sculpting can still occur anywhere on the model and the symmetrical mirroring will occur correctly.

3.3.8 Paint-on Resolution

In Future Work, section 3.2.11, the paper proposes the addition of a feature to allow a user to ‘paint on’ extra triangles where detailed sculpting was required. I subsequently investigated and implemented this facility. Users had found that the application as described in the paper did not accommodate enough triangles to allow intricate sculpting of small features. However, surface-wide re-tessellation results in an unacceptable reduction in performance, due to the large number of triangles produced.

This can cause vibrations in the haptic response because of a reduction in the refresh rate. It is often the case that there are areas where the triangulation is sufficient and others where it needs to be finer. A new tool type was added, which allowed the user to ‘paint on’ extra triangles. The tool has the appearance of a paint brush, but instead of colour, it applies a mist of yellow specks to the surface (nick-named “stardust”). The concept is that the user paints stardust over the areas that need finer detail, and these areas have their component triangles subdivided in a ratio of 4 to 1. This subdivision takes some time (typically a second or so) and so does not happen in real-time as the
user is painting, but rather after the action of painting the selected areas is complete and the user is ready to sculpt again. The subdivision is performed by dividing each selected triangle’s sides in half and interconnecting them to form four new triangles to replace the single former triangle. The algorithm then visits all lists in the system and makes any necessary additions and substitutions.

3.3.9 Performance

In section 3.2.11, Future Work, the paper mentions spatial indexing as something that could be pursued for further optimisation. I undertook this work after publication of this paper. During each graphics cycle, the system must find vertices that are close enough to the haptic tool tip to be likely candidates for collision with the tip in the next 33 haptics cycles. I have now improved the search for these candidate vertices by initially constructing a spatial index of all vertices. The spatial index structure is based on a grid of axis-aligned cells, filling the whole of the workspace accessible by the user’s haptic tool. This list is stored in memory as a linear array of cell objects. Each cell holds a linked list of its contained vertices. Since the cells are axis-aligned, and of a uniform dimension, they can be rapidly accessed from the knowledge of a 3D point in space, according to the equation 3.4.

\[
\begin{align*}
a &= \frac{(x - x_{\text{min}})}{x_{\text{dim}}} \times \text{num}_x \\
b &= \frac{(y - y_{\text{min}})}{y_{\text{dim}}} \times \text{num}_y \\
c &= \frac{(z - z_{\text{min}})}{z_{\text{dim}}} \times \text{num}_z \\
\text{index} &= (a, b, c)
\end{align*}
\] (3.4)

where \text{index} = the resulting 3D index into the array of cells

\[x, y, z = \text{the global coordinate of the point}\]
\[x_{\text{min}}, y_{\text{min}}, z_{\text{min}} = \text{the coordinates of the origin of the spatial index}\]
\[x_{\text{dim}}, y_{\text{dim}}, z_{\text{dim}} = \text{the spatial index x, y, z dimensions}\]
\[\text{num}_x, \text{num}_y, \text{num}_z = \text{the number of x, y, z cells}\]

At each graphics cycle, the haptic tool tip position is used to calculate an index into the array of cells, giving the cell within which the tool tip centre currently lies. However, depending on the cell size and the tool tip radius, the vertices that are candidates for collision may come, not only from the chosen cell, but also from a number of
surrounding cells. A list of index offsets to these surrounding cells is pre-calculated whenever the tool tip radius is changed by the user, according to the algorithm 3.5.

\[
\begin{align*}
\text{for all } i & \in -\text{rad} \leq i \leq \text{rad} \\
\text{slice}_\text{rad} &= \sqrt{\text{rad}^2 - i^2} \\
\text{for all } j & \in -\text{slice}_\text{rad} \leq j \leq \text{slice}_\text{rad} \\
\text{row}_\text{len} &= \sqrt{\text{slice}_\text{rad}^2 - j^2} \\
\text{for all } k & \in -\text{row}_\text{len} \leq k \leq \text{row}_\text{len} \\
\text{store index } (i, j, k)
\end{align*}
\]

where \( \text{rad} \) = integer cell index radius to be covered
\( i, j, k \) = integer cell index offsets in x, y, z dimensions

Candidate vertices for collision are therefore collected from the central cell, plus the spatially surrounding cells using the pre-calculated index offsets applied to the centre cell index.

This method avoids a global search for vertex collision candidates, but has the penalty of requiring that any moved vertices be reassigned to new cells. The trade-off was found to be positive, however, and resulted in an increase in the maximum vertices that could be smoothly sculpted by 50%. Using a Dell Precision dual processor 3.2 GHz computer running Windows XP, it was possible to smoothly sculpt a model comprising 22,528 triangles without any noticeable vibration.

Cavusoglu and Tendik used a method of passing linear approximations of a deformable surface to the haptics thread and updating the physical model in the slower graphics thread, to achieve the required performance in a simulation of deformable body organs [47]. My method improves on the accuracy of this by using the actual surface vertices at haptics rates. This is possible because of the optimisations described here and the fact that we are sculpting a surface ‘skin’ with no volumetric calculations being performed.

### 3.3.10 Inheritance Framework

Section 3.2.7 describes classes comprising the Deformable Shapes library. The C++ classes referred to are produced by sub-classing from suitable Reachin API nodes which are themselves classes in the Reachin API library. For example, the DeformableTriSurface class inherits from the Reachin API IndexedFaceSet node. (In this terminology, a node is a ReachinAPI class that can be built into a scene-graph. The...
terms can be used interchangeably in this context). The next task is to decide if any extra fields are required for the class. For example, the DeformableTriSet node has a *cohesion* field. This can be thought of as a ‘plug’ where a route can be connected from another field, or where a value can be set explicitly. The fields are added according to the pattern detailed in the Reachin API documentation. The next stage is to provide a VRML interface for the node. This allows the node and its field values to be specified within a VRML-style ASCII Reachin file. For example, in the DeformableTriSurface node this element allows the string “DeformableTriSurface” to trigger the creation of the node, and the string “cohesion” followed by a value to set that value into the field. The details of this mechanism are also described in detail in the Reachin documentation.

These haptic applications typically have two threads: one for graphics running at about 30 Hz, and the other for haptics control running at 1000Hz. Any operations that need to occur once per graphics update are added as part of the **Collider::collide()** function, that is a member of all nodes. In the DeformableTriSurface, the **collide()** function holds the algorithm for seeking out surface vertices that are nearby and candidates for collision within the next graphics cycle. If specialised graphics rendering is required, it can be added by overriding the **Renderer::makeGL()** function. If any special haptics are required, they are added by creating a specialisation of a ForceModel class and creating and installing it in the **collide()** function. The DeformableTriSurface ForceModel calculates the haptic reaction from tool-penetrating vertices.

Once these elements have been coded into the new nodes, the code is compiled and combined with other nodes into the DeformableShapes library, which can be imported into any VRML-style scene-graph file and used.

The artist’s workbench is an example of a virtual world containing objects with appearance and behaviour that is familiar to users’ experiences with similar objects in the real world. The next chapter discusses a method for implanting those virtual objects into the user’s current view of the *real* world.
Chapter 4.
Augmenting Reality

4.1 Introduction
As well as creating virtual objects that can be experienced in a virtual environment, it is possible to embed those objects into the real world using augmented reality. This paper, Haptic Collaboration with Augmented Reality, was written by Matt Adcock, Matthew Hutchins and myself and was presented as a poster at SIGGRAPH in August 2004. The paper appeared in the ACM SIGGRAPH digital proceedings for that year.

The paper describes an extension to the haptic virtual environment discussed in Chapter 3, to an augmented environment, where virtual objects are mixed with the real world objects, both of which can be seen, touched and interacted with.

Contribution: I developed the ‘Plausible Physics’ simulation referred to in this paper as well as the ability to grasp another user’s instrument haptically. These technologies are discussed in more detail in other chapters of this thesis. I also provided advice on issues concerning the interface between haptic code and ARToolKit [99]. It should be pointed out that, although ARToolkit forms a large component of the technology discussed in this chapter, I had no part in its development and it was not developed at my institution.

The work in our laboratory involved incorporating it into our haptic environment.

It should also be pointed out that I had only a minor role in this work. However, for reasons of completeness, and as it appeared at SIGGRAPH, it is a valuable inclusion in this thesis. Page limits prevented the opportunity to explore issues of software structure and design in this paper.

SIGGRAPH attracts delegates with highly technical expertise and interests in graphics, virtual reality, simulation, media and animation.
4.2 Publication 2: Haptic Collaboration with Augmented Reality

Matt Adcock, Matthew Hutchins, Chris Gunn
CSIRO ICT Centre, Australia
{matt.adcock, matthew.hutchins, chris.gunn}@csiro.au

Abstract

We describe a (face-to-face) collaborative environment that provides a coherent mix of real world video, computer haptics, graphics and audio. This system is a test-bed for investigating new collaborative affordances and behaviours.

4.2.1 Collaborative Haptics and AR

Our project goal is to develop a system that can provide a combination of haptic interaction, 3D computer graphics and auditory display to support collaborative tasks such as design, mentoring and data exploration.

The majority of haptic/graphic system configurations are designed for single user experiences. Usually they consist of a haptic device sitting next to a computer monitor. For a more co-located configuration, we have combined our PHANTOM haptic devices with stereo shutter glasses and a mirror [221] (see figure 4.1).

Collaborative haptic applications have now started to emerge [15] [84]. However, the majority of these systems have been targeted at tele-collaborative rather than face-to-face situations. We believe that Augmented Reality (AR) technologies are well suited to enabling haptic applications to be developed to support and enhance face-to-face collaborative tasks.

4.2.2 The CSIRO Collaborative Haptic Toolkit

The Reachin API (formerly Magma) [221] was created in order to better integrate haptics and graphics rendering. It is based on a haptic/graphic scene-graph that borrows a lot of its structure from VRML.

Last year at SIGGRAPH, Gunn et al. [84] described a Collaborative Haptics Toolkit extension to the Reachin API. This Toolkit contains a ‘Plausible Physics’ simulation that mediates interactions between users and objects and also direct interactions between users.
In an example surgical training application, an instructor can ‘grasp’ the student’s tool to haptically guide it while the student feels the force of the instructor’s guiding hand. Similarly, the instructor feels any resistance caused by the student. They can also collaboratively push, stretch, pull the objects (organ and tools) around the scene, with attached objects stretching and moving accordingly. They converse through a separate voice link.

A problem arises when using two Haptic Workbenches are in the same room: the two participants are still visually ‘cut off’ from each other and they cannot use any unmediated communication except for the occasional shouting between workbenches.

4.2.3 Haptics with the ARToolKit

The ARToolKit [99] allows developers to include vision-based marker tracking in their applications. It is primarily used to render virtual objects in such a way that they are perceived to be co-located with specific fiducial markers. Inversely, it is also capable of tracking the camera location with respect to the marker.

We have developed a new Reachin-compatible node that encapsulates the ARToolKit routines needed for camera tracking [4]. This allows us to ‘break free’ from the workbench configuration. Figure 4.2 shows an example in which we use a marker on the desk as a reference for the placement of the virtual objects. Here, we have offset the two real haptic styli from the virtual ones.
4.2.4 Conclusion

We have created a collaborative environment in which users can interact with virtual objects and each other, while maintaining most of their unmediated communication abilities. This system is acting as a test-bed for investigating collaboration affordances and behaviours.

==================================================================
4.3 Discussion

The paper discusses a method of haptic collaboration suitable for two, co-located users, allowing them to interact directly with each other and simultaneously with virtual objects that are perceived by both to be in their shared local environment. The collaboration described here should not be confused with the chapters on collaborative haptics, appearing later in this thesis. The latter refers to the ability to allow two users to interact haptically using a network connection between two computers over a distance. This paper refers to the ability to allow two co-located users to share a virtual 3D model from different viewpoints and to mix this with real video of their surroundings.

Because of page limits, the paper did not go into any detail about the haptic interaction with the model and its relationship with the fiducial marker. The ARToolkit API uses analysis of the pattern in the fiducial marker to provide both the position and orientation of any marker it identifies. It then binds the position and orientation of the virtual model to this (sometimes with an offset). This allows a user to move the fiducial marker (which in our case was a printed pattern on a sheet of polystyrene) around on the table, or even hold it in their hand, and see the virtual model move with it. The authors’ contribution involved encapsulating the ARToolkit code into a node of the haptic scene. In this way, the ARToolkit node could control the position and orientation of the virtual, haptics-aware objects in the scene. If the whole virtual scene, including the haptic tool position, is bound to the fiducial marker, then the model can be touched and felt wherever it is placed. If only the model is bound to the marker, the haptic tool position remains fixed, allowing the model to be brought up to within touching distance of the haptic tool for haptic interaction. However, in this latter configuration, the ARToolkit image tracking system is not able to fix the virtual model rigidly enough to allow usable haptic interaction. The slight jitter in positioning causes unpleasant vibrations with the haptic tool.

The system provides an intuitive ‘left hand / right hand’ technique of interaction with a model, where each hand performs a different, but complementary action, for example positioning and indicating [111]. It is therefore possible to allow any of the haptic applications and demonstrations described in this thesis, to be run in an augmented reality mode, by including the ARToolkit node into the scene and making the appropriate field network connections.
The ability to see and touch virtual objects in the same environment as real objects can be a very powerful tool in a learning environment. Since the ARToolkit API allows the separate recognition of a number of different fiducial markers within the view of the head-mounted video camera, it is possible to load in different virtual models as each independent marker is identified by the system. Although the system as described was built with only one fiducial marker and one associated virtual model, it is conceivable that multiple models could appear at different times, or together, as different, unique markers appear within the camera’s field of view. Such a scenario could involve one or more students following printed reading matter and having a different virtual model appear when a new marker on the page is recognised. They could then interact with the model to learn and discover its behaviour and properties. The interaction can take the form of touching and manipulating the virtual objects as if they were real – grasping, holding, feeling the weight and pliability. Alternatively it may take the form of exploring the structure of the model through ‘super-real’ interaction – for example using transparency, moving cross-sectional slices or labelling components.

In the anatomical example used in the prototype, the users can refer to an anatomy text on the desk and also feel the pliability of the virtual body organs. As well, they can shrink their viewpoint down to a miniature size and fly through the body, or make organs transparent to see and feel the structures behind them. This capability opens up the prospect of a wide variety of experiential learning situations involving text, diagrams, videos as well as interactive, touchable models. The authors developed a prototype of such a system which involved the exploration of the human scull and brain. The system used transparency to allow users to see through the scull and touch the brain, which deformed, approximately, to the touch. They could also then select either an MRI 6 slice or photographic slice 7 which could then be pushed through the head, displaying the varying image-based data in-situ with the 3D model. This prototype is able to be run within both the CSIRO Haptic Workbench (described elsewhere in this thesis) and in the augmented reality configuration.

It is possible to combine both real and virtual models in a learning exercise. A plastic model could include a fiducial marker that activates a virtual addition to its structure.

6 Magnetic Resonance Imaging
7 From the Visible Human Project [156]
An example of this would be a plastic skeleton that may initially be covered by a virtual body. Touching the body with the haptic tool could trigger the removal of the skin to reveal the muscles and tissue. This could also be removed to reveal the vascular system, and so on. Finally the haptic tool could touch the real plastic of the skeleton model, with very little discrimination between touch associated with real objects and with virtual objects.

As the paper mentions in its conclusion, the users still have most of their unmediated communication abilities – they can see and talk to each other and are aware of changes in their environment, such as people coming into the room, telephones ringing etc. However, their communication ability may be somewhat diminished by the unwieldy head mounted displays that are used. The resulting lack of eye contact may impinge on the effectiveness of face-to-face interaction with others. Improvements in the HMD technology, especially a reduction in the size and weight, will mitigate this problem.

The paper mentions that two co-located users participating in a shared virtual environment (not augmented reality) on separate work stations “cannot use any unmediated communication except for the occasional shouting between workbenches”. Since this paper was written, we have conducted trials involving two users working in such a collaborative virtual environment, with electronic audio and video communications, but being both located in the same room. This configuration can result in a ‘doubling’ of the audio, with the users hearing each other’s voices directly (through the air) as well as via the audio channel of the system. The encoding and decoding of the audio system results in a temporal mismatch compared to the direct audio, so that the users hear each other’s voices twice. In the trials, this proved to be quite distracting. One solution was to move the workstations a sufficient distance away from each other to diminish the air channel sufficiently as to be unobtrusive. If the electronic audio/video channel is dispensed with, we have the situation referred to in the paper, where the users may need to raise their voices to communicate properly – especially in an environment with any noise. The augmented reality system described in this paper solves this issue by allowing the users to be positioned close to and opposite each other, looking on a common work-space (typically a table) and interacting together with both the real objects and virtual objects. The close proximity, and the face to face orientation with no intervening barriers or equipment, encourages natural conversation at a normal volume.
This work used haptic objects which are familiar from real world experience. The next chapter investigates how haptic technology can extend beyond this into the unreal effects that can aid in performing a task.
Chapter 5.

Extending Reality

5.1 Introduction

The previous two papers investigated using haptics to create virtual versions of real world objects (e.g. clay) in both a totally virtual environment and an environment that allows them to be mixed with real objects. This paper extends the technology to also include ‘fictitious’ haptic artefacts which have no direct correspondence in the real world, such as guiding force fields. The combination of these technologies, then allows the inclusion of real entities, virtual simulations of real entities and virtual ‘fictitious’ entities within a single working environment.

The paper, Experiments in the Haptic Rendering of Constraints: Guiding the User, was presented at the Simulation Technology and Training Conference in Melbourne in 1999.

Contribution: I wrote the paper, devised the experiments and developed the algorithms for this work. My co-author assisted with some of the implementation. The paragraph entitled Data Effects (page 71) mentions separate research undertaken by others in CSIRO, in which I had a minor involvement.

The paper discusses using extra forces, outside those that might be employed to simulate reality, to assist users in some task. Whereas developers of simulation applications generally strive to approach reality, this paper describes techniques which are deliberately intended to introduce features that are not present in reality, and hopefully achieve outcomes that supercede those possible in a real environment in some aspects. This technique is sometimes referred to as providing virtual fixtures. However, this term implies using fixed artificial guiding surfaces. The paper presented here
addresses a more general technique - that of providing force fields that can vary in strength and direction depending on the user’s actions.

The conference’s audience comprised representatives of government departments, the military and medical administrators and practitioners, all with an interest in using virtual reality for training. There were only a small proportion of technical personnel and as a consequence, the paper does not go into any depth on software structure and design. Most attendees had no knowledge of the use of haptics in a virtual reality environment. The paper introduces haptics and describes the application as a whole without pursuing specific issues concerning haptics. The paper described in Chapter 6 covers a similar area of work, but addresses an audience familiar with the technical aspects of the topic, and as such focuses more on the haptic implementation issues.
5.2 Publication 3: Experiments on the Haptic Rendering of Constraints: Guiding the User

Chris Gunn,
Paul Marando
CSIRO, Cooperative Research Centre for Advanced Computational Systems (ACSys)
RSISE Building (115), Australian National University
CANBERRA, ACT, 0200

Keywords: Haptic, Virtual Environment, Virtual Reality, Immersive

Abstract
This paper describes some experiments that we have conducted into using haptic (force) feedback to influence the actions of a user of a 3D design tool. We are researching the potential of using a co-located hapto-visual virtual environment, called a Haptic Workbench, to significantly improve the efficiency of "free form" 3D design. The concept behind this form of user interface is to try to create a "fuzzy UI", in which the user's actions can be guided by applying a force representing certain rules, but the user is not obliged to comply with those rules.

This is in contrast to the traditional computer user interface, which often renders the constraint graphically and then uses a 'beep' and an error message to inform of the rule violation. The benefit of this ‘fuzzy’ user interface is that the user is aware of the rule transgression, but still has the capability to make the decision to continue on, by pushing harder against the applied force.

We also used the system to overcome problems of user fatigue and an unsteady hand when drawing in 3D. Various experiments were performed that related force constraints (both attractive and repelling) to curves and surfaces in 3-space.

5.2.1 Introduction

Haptic Feedback
Virtual environments often involve a 3 dimensional pointing device. Often, this can be waved around in space, with its 3D position being tracked electronically. Haptic feedback, in this context, involves applying computer-controlled forces to this pointing device, typically by an attached mechanical arm. At the Cooperative Research Centre for Advanced Computational Systems (ACSys), Virtual Environments Laboratory in
Canberra, we have been investigating the use of haptic feedback as a means of applying guiding constraints to a user in a computer aided design (CAD) application.

5.2.2 Background

We are using an immersive virtual environment that we have named a Haptic Workbench, described in 1998 by Stevenson et al. [221]. It is an extension of a Virtual Workbench developed by the institute of Systems Science at the National University of Singapore, described by Poston [173] in 1993.

To produce a 3D effect, the user wears polarising shutter glasses, synchronised with a left-and-right image displayed alternatively on the screen at 60 hertz per eye. The left and right eye convergence point is set such that the model being viewed appears to be behind the screen. This, by itself, produces a 3 dimensional effect, but does not provide any immersion. To achieve this, the computer monitor is mounted on a frame, and pointed vertically down onto a mirror, mounted at approximately 45 degrees. The user then looks at the mirror, not the monitor. This means that they can now place their hands behind the mirror, exactly where the model appears to be. With their hands in that position, they can now pick up and use two tools. The left-hand tool is tracked in 3D space by a Polhemus tracker. Typically it moves and rotates all the objects in the scene.

The right-hand tool is a haptic feedback device, known as a PHANTom, manufactured by SensAble Technologies [201]. It consists of a probe connected to a multi-linked mechanical arm. The arm has internal cables driven by sensitive electric motors. It provides force feedback to the user’s hand as they move the probe in the space below the mirror. Software was written to merge the graphic and haptic coordinate systems, resulting in the co-location of both representations of the model. A C++ class library, Magma [221], was also written to enable shapes to be treated as single entities, without having to worry about the separate haptic and graphic representations. The user can see the objects in 3D and ‘feel’ them through the haptic tool.

Figure 5.1: The Haptic Workbench, showing PHANTom haptic device.
The haptic tool can be animated in the scene as any type of simulated tool; often a fork or a pencil, and can perform work on the objects in the scene.

### 5.2.3 Haptically Assisted Drawing

The use of force to simulate solid objects enhances the reality of the model being observed. However, haptics can be used for purposes other than simulating reality. It would be advantageous to use the Haptic Workbench to create new models, not just explore existing ones. To do this, we must allow the user to draw objects in the space behind the mirror. We have noticed that users find it difficult to draw in a 3 dimensional space, with nothing to press against or guide the hand, and arm fatigue easily sets in after only a few minutes of sketching. We decided to investigate providing haptic constraints to assist in free-form drawing.

### 5.2.4 Fuzzy Feedback

We then extended this concept to provide a type of ‘fuzzy’ feedback. A conventional computer user interface usually provides feedback to the user in the form of an error message and a ‘beep’. This gives only a binary form of feedback; either you’re right or you’re wrong. Even a warning message, which does provide a form of advice without enforcing an action, is disruptive to the flow of work, and usually has to be acknowledged. It does not provide the ability to continuously vary an influence on the user; to guide and advise continuously as they are working.

Haptics, however, provides just this ability. We can introduce a gentle force, guiding the user’s hand in a certain direction. We can vary the strength and direction of that force as the pointer is moved, giving the ability to smoothly vary the amount of ‘advice’ we are sending. This is a fuzzy feedback mechanism. The user can easily over-ride the force by simply pressing harder against it. Our trials have shown that it is intuitive and immediate. It doesn’t interrupt the flow of work and mimics many real world situations.

### 5.2.5 The Problem Domain: Underground Mine Planning

We investigated several types of constraints by building a prototype application based on the underground mining industry. A particular problem with the planning of an underground mine is that of drawing the proposed access tunnel. This is often referred to as the ‘drive’ or ‘decline’, and is generally a road that slopes down to the ore bodies that are to be mined. Since this is a 3 dimensional drawing problem, the 3D immersive view is of immediate benefit for producing the initial sketch. However, there are certain
restrictions on the route that the decline can take. Obviously it should pass close to, but not necessarily through the valuable ore. It should also avoid any known dangerous areas, such as water logged or unstable rock. It should have a limited gradient, determined by the capabilities of the vehicles that are to be used. Any bends should be of such a curvature that the vehicles can negotiate them. The route should be as ‘smooth’ and short as possible, minimising twists and turns. There may be an outer boundary, perhaps introduced by the limits of the mining lease. These requirements may well conflict with each other and require some expert knowledge, in the form of user interaction, to make decisions between different options.

This problem domain seemed a suitable one to investigate a haptically constrained CAD system. We have written a prototype application allowing a user to sketch a line in 3-space around some solid objects representing geological entities. The application has a series of haptic buttons with which the user can turn on or off various types of constraint. Each constraint type also has a slider to modify its intensity. The haptic force on the user’s sketching tool is the resultant of all forces currently turned on. A visual meter in the bottom left corner displays the magnitude of each component of the force, allowing the user to glance at this if they are unsure which particular constraint is dominating at any one time. The bottom left corner of the scene has a 3D ‘compass’ that rotates with the user’s left hand tool. This was found necessary, as, when the user rotates the scene it is very easy to forget which is up/down/east/west etc.

5.2.6 Types of Constraint

The following types of haptic constraint are being explored:

**Force repulsion from objects.**

To satisfy the requirement of guiding the user away from certain areas (e.g. dangerous zones), we implemented a repulsive force from nominated solid objects in the scene. The idea is that the user can ‘feel’ the danger, and be guided away. However, they may be constrained by other effects, and have the option to proceed further, by pushing against the force.

The objects were rendered haptically and graphically, so that the user can both see and ‘touch’ them with the haptic tool. Initial versions used simple spheres as the objects, as they were the easiest to model. A slider could set a surface repulsion force and a maximum force distance. The force produced was a linear function between these two.
In use, the effect is much the same as when you bring similar ends of two magnets together. The tool is repulsed away or guided around the object. However, by pushing harder, the user can draw right up to the object.

The spheres were then replaced by triangulated surfaces, allowing a more accurate representation of a scene. The triangulated surfaces themselves are haptically rendered as hard surfaces, so that, with no repulsive force turned on, the user can still ‘feel’ them when the tip of the tool collides with them. The algorithm for collision detection uses a binary spatial partition (BSP) tree, allowing a fairly efficient search for triangles (if any) with which the tool is colliding.

If we are to model a force emanating from the closest triangle on the surface, we must search all triangles, to find it. This increases the cycle time greatly. It is a feature of haptic feedback devices that the refresh time must be kept very short - of the order of 1 millisecond - to avoid feedback oscillations and sudden jolts. Such jolts can be potentially so violent that they could be dangerous to the user, and it is common to have a safety cut-off if the refresh rate gets below a certain amount. We discovered that with a simple search of all triangles to calculate a repulsive force, we could not scan many triangles on an SGI Octane before exhausting the 1 ms refresh time. Since this is clearly not satisfactory, we are developing a level-of-detail system incorporating a hierarchy of bounding spheres. When the user’s tool is a long way away from an object, the repulsive force is calculated as if the object were a sphere. As they get closer to the object, the next level in the hierarchy is activated and they ‘feel’ a force emanating from two sub-spheres centred about a sensible division of the object’s triangles. As they continue to approach, they progressively go to smaller spheres until the lowest level is broached and they are exposed to the actual triangles within the innermost sphere.

This algorithm has the added benefit, that the number of hierarchy levels can be adjusted to the capabilities of the hardware platform. In this way, it will also allow users to improve the accuracy of the application as faster processors become available.

**Gradient and curvature control.**

There is often a requirement to draw a line at a certain gradient. Such is the case when drawing the mine’s decline.

We have experimented with inserting a haptic cone under the drawing tool. This, however, does not prevent the user doubling back along the cone’s surface. It also does
not allow them to change direction and keep the same gradient (they would ‘launch’ off the cone). We then experimented with calculating the gradient of each drawn line segment. If it was beyond the limit, a force was applied guiding the user back to the limit. Optionally, the erroneous line segment can be corrected automatically. A similar, running calculation can be applied to curves, by measuring the curvature of the last two line segments.

**Line following**

When a haptically rendered model contains lines, it is not obvious how to handle the haptic properties of the line. After all, a rendered line is very thin, and is therefore extremely hard to ‘touch’ with a haptic tool. We found that a ‘suck-in’ effect, drawing the tool to the line if it was in the vicinity, was quite intuitive, and as far as some users were concerned, they could see no inconsistency between this and the surface modelling in other parts of the haptic scene. This technique was used successfully in an application that required the user to follow a single ‘wiggly’ line amongst a few hundred. This application was an instance of using haptics for data mining. The data was a series of aeromagnetic traces, recorded from a plane flying a grid over an exploration lease. The user was looking for irregularities in the magnetic recordings from each flight line.

**Sketched line repulsion**

We extended the idea of repelling the tool from a solid object, to repelling it from a line. This line could, of course, be the same line that is currently being drawn, guiding the user away from crossing over their own track.

**Treacle**

It was noticed that a user drawing freehand in 3D space would often trace an erratic path. We overcame this by introducing a variable damping force, which we nicknamed ‘treacle’ as it feels like you are drawing in treacle. This made a profound difference in the drawings, producing much smoother controlled curves.

**Plane drawing**

At times it is necessary to draw a 2D drawing in a 3D scene. That is, to draw on a plane within the overall scene. We are allowing for the creation of a temporary plane that is both visible and haptic. It can be moved with the left-hand Polhemus tool, and pressed against by the right hand haptic tool, much as you might hold a clipboard to draw on it. When completed, the plane is removed, leaving only the drawing. This idea should be
general enough to be applied to any surface, so, for example, you could draw onto a sphere.

**Continuation of line**

A predictive algorithm is being researched, which will make the easiest path for the tool a continuation of its most recent path. For example, if the tool has been following a straight line, the path of least haptic resistance will be the continuing straight line. If the tool has been following a curve, continuing the same curve will meet no resistance, while all other trajectories will encounter some force.

**Boundary**

A variation of the repulsion from an object is the situation where the tool needs to be inside a volume and must be prevented from venturing outside it. We are implementing this as a series of bounding surfaces with repulsion. This introduces the problem of how to turn the boundary on and off if the control widgets are not within the boundary (See ‘Control Widgets’, below).

**Data Effects**

Certain types of data may be better interpreted if they are mapped to a haptic force. We developed an application to interpret 3D seismic data, with the auto-correlation of the data being mapped to a haptic effect. This produced an effect where features of the data felt like ridges and troughs. Other aspects of the data were rendered graphically. The user could then easily move around the ‘troughs’, perhaps sketching in some annotation.

**5.2.7 Problems:**

**Oscillation**

We have found that some situations produce unwanted oscillations of the haptic drawing tool. This can occur, for example, when there is a gradient limit on both positive and negative gradients. We succeeded in minimising this through damping.

**Process Time**

The haptic calculations must take place in approximately 1 millisecond. This is because, unlike graphics, if the haptic rendering is slow, the tools can get vibrations and behave erratically. It may even be possible to move the tool into an ‘impossible’ position between cycles and have it violently thrown out when the update finally occurs. (Most systems have an automatic cut out, which crashes the program before any damage
happens to the hardware or the liveware!). It is necessary, therefore, to keep algorithms small and efficient, and this is currently one of the major challenges facing us.

### 5.2.8 Control Widgets

As described above, there are many types of haptic aids to drawing that may be required during a particular task. The user may want to switch these on and off as they work. We have provided buttons and sliders at the bottom of the scene for this purpose, but it may not always be convenient to move the tool from the drawing task to the button array and back again. We are investigating two solutions to this problem. Firstly, we can use the twisting ability of the haptic tool to make a selection from a radial menu displayed around the tool. Eventually we would like to incorporate some speech recognition software to allow the user to issue commands orally while keeping both hands on the drawing task.

=============================================
5.3 Discussion

The paper states that “our trials have shown that it is intuitive and immediate”, when referring to using haptic effects to pass advice to a user. At the time of writing, these trials were ad hoc demonstrations of the software to colleagues and visitors to the laboratory. Chapter 7 applies a more rigorous investigation of this issue, reporting on some experiments that were conducted to measure any assistance that the technique provides.

The paper uses the phrase ‘fuzzy feedback’. This is derived from the term ‘fuzzy logic’ and refers to feedback that portrays gross, approximate information. Sheridan [208] explains this use of the word ‘fuzzy’ as:

“the observations and thoughts of most people most of the time may be said to be mentally modelled and/or communicated to other persons in terms of sets of natural language words (such as ‘short’, ‘medium’, ‘long’ and ‘very long’ as applied, for example, to a train, or ‘urgent’, ‘little’, ‘normal’, and ‘plenty’ as applied to time available for a train to meet its schedule) ……..Precise quantification and Boolean logic imply ‘crisp’ sets, wherein a number or symbol is identified with a well defined set of objects or events, and explicitly not identified with other objects or events” (page 69).

My use of the word in the context of haptic feedback reflects this. The user is not meant to interpret the exact value that the feedback is representing, but instead recognise the general message that is being conveyed. They can then decide whether to fully comply with the advice, fully ignore it, or take a compromising course in between.

One of the benefits of using a haptic channel of information flow between a computer and a user is that it does not require any visual representation. Often when using a graphical computer interface, the visual sense can be overloaded with information [20]. Such ‘screen clutter’ can lead to confusion, fatigue and missed information. Supplying information via an audio channel can be a valid alternative, but this also has a limited information capacity. The haptic channel, investigated here, is a third channel which can be utilised to advantage by spreading information over alternate feedback mechanisms. A force is a vector, comprising both magnitude and direction, but it can only portray a single piece of vector-information uniquely. It is possible to combine multiple vectors into a single resultant force, but this disguises the relative contributions of the components. In some circumstances it may only be the total resultant vector that
is important, but it is problematical to provide information on what are the separate causes of the haptic message being provided, if indeed that is needed.

If we were to place complete trust in the computing system to direct our motions, then it may well be acceptable to comply without question to a resultant guiding force that is a combination of a number of components, all originating from separate, pre-programmed rules. However, if this were so, then why do we need a human controller at all? Presumably the human contributes some expertise that the computer system cannot provide (for example, knowledge of geological structures, faulting characteristics and chemical occurrence in close proximity); otherwise the whole solution could be done automatically. Therefore it can be argued that, if a guiding force is to be supplied to a user, it needs to be easily identifiable as representing the result of a single rule, and if not, then other indicators, such as on-screen visuals, need to be provided to allow the user to gauge the relative contributions of the various components making up the whole. Obviously, this would then consume some of the screen real estate. Care would need to be taken in the design of any application to ensure that the extra expense of providing a haptic capability did indeed solve the particular problem addressed.

In the prototype described in the paper, a visual meter is shown in the bottom left corner displaying the magnitude of each component of the force. This allows the user to glance at this if they are unsure which particular constraint is dominating at any one time. However, this solution to the problem obviously adds to the screen clutter, one of the reasons for using a haptic solution in the first place.

The experiments detailed in Chapter 7 show that a design which does successfully display a force representing a number of components is still possible, as long as the components are the combination of rules of the same class, i.e. representing simultaneous violations of the same rule in more than one location.

In the section entitled **Line Following** (page 70), the paper mentions that the use of an attractive force towards a line or curve in space can assist a user in following the line with a pointing device. Such techniques were investigated in [122], where the aim was to provide a learning aid using haptics. They found that there was no greater learning benefit in using haptics in this way, over the use of graphics alone. My paper is not addressing the issue of haptic learning aids, but instead addresses the use of haptic aids to assist a user in a precise positioning task, *during the task itself*. It applies haptic attraction to allow a user to select one of hundreds of similar, closely spaced lines, and
to listen to an audio display of some spatial variable distributed along the line. The users could ‘touch’ the line (by being snapped onto it from close proximity) and then slide their haptic tool along it, listening to a sonification of the aero-magnetic data that had been collected from a grid of aircraft flight paths. The purpose was to detect irregularities in the magnetic recordings that the data represented. The users had the ability to quickly scan a line but also concentrate on a small section, carefully passing the tool back and forth across a region of interest. Since the lines were narrowly spaced, erratic and sometimes crossed, the haptic effect was seen to be a particularly useful and intuitive way to select and ‘stay with’ one particular line, as well as to break away from it and select another when desired. A most interesting HCI issue was that the users reported no inconsistency between snapping to a line and ‘touching’ a virtual surface. As far as the haptic algorithms are concerned, these effects are the opposite – in one case there is an attraction towards a point (the nearest point on the line) and in the other there is repulsion from a point (the point of penetration of the tool beneath the surface). However users in both cases reported ‘touching’ the object, and seemed to notice no inconsistency in the interface. This is significant because, in the real world free standing lines as such do not exist, at least in a physical, touchable sense. A line in a virtual world is infinitely thin. Its visual representation may be one pixel in width, but its logical width is zero. A haptic representation of an infinitesimally thin line, or even a line of one pixel’s width, is problematical without taking advantage of the illusion reported here.

The section entitled Plane Drawing (page 70) mentions that the technique of providing a virtual hard surface to assist the hand when drawing a line on a plane “…should be general enough to be applied to any surface, so, for example, you could draw onto a sphere”. Since the publication of this paper, I have implemented this capability, allowing the drawing of a 3D line on the surface of any object, for example a plane, sphere or irregular shape. Optionally, the object can be removed to leave the 3D line ‘floating’ in space. This is reported in Chapter 3 of this thesis.

In the section entitled Continuation of a Line (page 71) the paper mentions that “A predictive algorithm is being researched, which will make the easiest path for the tool a continuation of its most recent path”. Subsequent work towards this goal proved to be unsuccessful. The aim was to predict the user’s intentions when drawing a freeform line, and to provide some ‘just-in-time’ haptic guidance to assist them to achieve their
I developed software to track the user’s actions and apply forces to simulate transient ‘grooves in space’ that would assist them to draw more smoothly and accurately. As the user sketched in 3D, the system would analyse the succession of line segments and attempt to fit them into one of three classifications; straight line, curve or spiral (a curve of constantly changing radius). It proved to be very difficult to get an acceptable balance between timeliness and correctness. If the system was tuned to predict the user’s intention early enough in the motion to be of any benefit, it would often categorise their path incorrectly, and apply forces which were divergent from those required to assist the user’s true intention. Alternatively, if it was tuned to delay the prediction until a sufficient amount of the user’s path was detected to predict the intention with more certainty, it was often too late to be of any assistance, as the user had typically already completed most of the required motion. The task was complicated by the fact that the freeform drawing activity was taking place in 3D, with a haptic tool held, unsupported in space. The complexity of the prediction algorithms was increased by the three dimensionality of the user’s motion [149], as the user was drawing lines that could curve in any or all of the three dimensions. Added to this was the unsteadiness of a user’s hand because they had no surface to rest on [62]. This resulted in an unintentional variance in any drawn line that was difficult for the algorithms to accommodate. Other work in this area [63] has had more success using visual predicted guides, as opposed to haptic ones, to assist freeform 3D drawing.

5.3.1 Force calculations

The various types of haptic user feedback employ different algorithms to calculate their forces. The repulsion from an object is described in the section, **Force Repulsion from Objects** (page 68). This refers to a hierarchy of spheres containing a triangulated object. When the user’s tool is outside of the outermost sphere, it feels a force of repulsion from the centre of that sphere. The force vector is calculated according to equation 5.1:

\[
force = k \ast (\vec{p} - \vec{c})
\]

where \(k\) = stiffness constant
\(\vec{c}\) = centre of the sphere
\(\vec{p}\) = tool position

(5.1)

When the tool penetrates this sphere, it is replaced by the resultant of the forces of repulsion of the enclosed spheres, each calculated according to equation 5.1. This creates a discontinuity of force at the penetration point. However, since these forces are
being used to provide the user with advisory feedback, and are not intended to be representing any real world object, it is acceptable for them to be quite weak. The low strength of the forces partially masks the discontinuity, so that it feels like a barely discernable ‘pop’ or ‘click’ when a sphere boundary is being traversed.

The intention of the hierarchical enclosing force spheres is to allow whole objects to contribute to the force on the tool without suffering the performance penalty of searching for triangles on each of the objects. When the tool is sufficiently far from an object, an approximation for the surface repulsion is used. Once the user’s tool penetrates the innermost hierarchical sphere of a particular object, the repulsion force is replaced by one from the closest point of that object. The object’s vertices are sequentially searched until the closest is found and then equation 5.1 is again applied using that point instead of a sphere centre point. This is a gross approximation to repulsion from the surface itself, especially if the triangles are elongated or the surface has individual spikes. However, it works sufficiently well for a convex or near-convex object.

The forces used to hold a drawn line to a gradient is calculated by computing the gradient of each segment of the line as it is drawn. If the gradient exceeds a threshold, a force is applied to the tool opposing the error. The calculation of the force is shown in equation 5.2. The gradient of the most recent segment is calculated. If it exceeds a threshold, the excess is used to provide a correcting force for the next segment.

\[
\text{seg} = \overrightarrow{ptB} - \overrightarrow{ptA} \\
yErr = \left( \frac{\text{seg}.y}{\text{seg}} \right) - \text{gradMax} \\
\text{if } |yErr| > 0 \implies \text{force} = (0, k \times yErr, 0) \\
\text{else } \text{force} = (0, 0, 0) \\
\text{where } \overrightarrow{ptA} = \text{drawn line segment start point} \\
\overrightarrow{ptB} = \text{drawn line segment end point} \\
\overrightarrow{seg} = \text{last drawn line segment} \\
\text{grad} = \text{the maximum allowed gradient} \\
yErr = \text{gradient excess} \\
k = \text{scaling factor}
\]
Curvature forces are calculated by measuring the angle between successive line segments and comparing this with a maximum allowed angle. The force applied is calculated to be in the plane of the two line segments according to equation 5.3.

\[
\begin{align*}
\text{segA} &= \overrightarrow{ptB} - \overrightarrow{ptA} \\
\text{segB} &= \overrightarrow{ptC} - \overrightarrow{ptB} \\
\text{midPt} &= \frac{2}{\text{segB}} (\overrightarrow{ptC} - \overrightarrow{ptA}) \\
\text{force} &= k \cdot \left( \overrightarrow{ptB} - \overrightarrow{midPt} \right)
\end{align*}
\]

(5.3)

where \( ptA \) = start of previous line segment

\( ptB \) = end of previous line segment and start of last line segment

\( ptC \) = end of last line segment

\( k \) = scaling factor

The curvature correction forces were not found to be useful or intuitive. They either over-corrected or came into play too late. This is due to the algorithm’s dependence on the previous two drawn segments before applying any corrective force.

The line repulsion forces were more successful. They used the algorithm shown in 5.4.

at each graphics cycle step
find closest point on line, closestPt
\[ \text{force} = k \cdot \left( \text{toolPos} - \text{closestPt} \right) \]
where \( k \) = force scaling factor

(5.4)

The line attraction algorithm is similar, with the direction of force reversed. However it was found that the algorithm shown above produced a rough stepwise varying attraction to each line point as the tool moved along the line itself. To overcome this, the force is modified by removing the component that is parallel to the currently touched line segment, according to equation 5.5.

\[ \overrightarrow{fPerp} = \overrightarrow{\text{force}} - \left( \overrightarrow{\text{force}} \cdot \overrightarrow{\text{seg}} \right) \frac{\overrightarrow{\text{seg}}}{|\overrightarrow{\text{seg}}|} \]

(5.5)

where \( \overrightarrow{\text{force}} \) = the force calculated according to 5.4

\( \overrightarrow{\text{seg}} \) = the current closest line segment

\( \overrightarrow{fPerp} \) = force perpendicular to line segment
The ‘treacle’ effect was created by supplying a force which is proportional to the velocity of the haptic tool but in the opposing direction, as shown in equation 5.6.

\[
\text{force} = k \cdot \frac{\text{prevToolPos} - \text{currentToolPos}}{\text{currentTime} - \text{prevTime}}
\]

where \( k \) = force scaling factor

\( \text{prevToolPos} = \) tool position in the previous haptic cycle

\( \text{currentToolPos} = \) tool position in the current haptic cycle

\( \text{prevTime} = \) the time of the previous tool position

\( \text{currentTime} = \) the current time

This work looked at a range of possible haptic effects that can be used as aids in performing a dexterous task. The next stage of the work delves deeper into one of those techniques.
Chapter 6.

Force Fields

6.1 Introduction

The previous paper (Chapter 5) introduced the concept of providing forces in a virtual environment that do not correspond to any real world objects or forces, but which may still be beneficial to a user performing a task. It was aimed at an audience of varying technical expertise and provided a high level overview of some of the types of forces that can be employed, but did not discuss them in any depth. This paper takes one particular case for the use of such forces, and presents the approach to an audience that was more technically based and familiar with virtual reality.

I presented this paper, Using Force Fields as a User Interface Device, at The IEEE Virtual Reality Conference (VR2005) in Bonn, Germany in March, 2005.

Contribution: 100%. I developed the concepts, and designed and implemented the algorithms and was sole author of the paper. The paper covers some of the same material as the previous paper in Chapter 5, but goes into more depth on the haptic aspects of using force repulsion from objects in a mine planning scenario.

This was an international conference with delegates from highly technical backgrounds, familiar with many of the issues relating to virtual reality. All were aware of the haptics field and many had direct experience with haptics. The conference was much more directed at the system and software issues than that addressed in the previous chapter.
6.2 Publication 4: Using Force Fields as a User Interface Device

Chris Gunn  
CSIRO, ICT Centre, Australia  
E-mail: Chris.Gunn@csiro.au

Abstract

This paper discusses the use of force fields as a means of passing advice from a computer program to a user. It is suggested that this mechanism can be used as a replacement for traditional user interface pop-ups or other widgets when the task is a three dimensional one and a haptic device is available. The benefits and limitations are presented, along with a description of an example prototype application in the underground mine planning domain.

6.2.1 The Problem

During the course of an interactive computer program, there are typically times when the program logic determines that some information needs to be passed to the user. The simplest case of this is the pop-up message box, often advising the user that they have violated some rule. In a 3D application, these messages may take the form of “you are too close”, “too far”, “and too steep”, “too thin” etc. The user typically needs to pause what they are doing and acknowledge the message, usually by clicking on a button. In a traditional, 2D application this can be disruptive to the flow of work. In a 3D task, the disruption can be greater because of the need for the user to move their interaction tool in 3D from their work location to the pop-up. If they were grasping, drawing or pointing at something, they may need to abandon their interaction to react to the message. Controlling a 3D interaction device is inherently more difficult than a 2D one (such as a mouse) because it lacks the stability and friction of a surface to rest on. Returning to the point of interaction after moving away can therefore be more difficult than with a 2D mouse and cursor.

The flow of information to the user may not just involve binary messages. Continuously varying data may also need to be conveyed to the user. An example of this might be the distance between two movable objects. A visual user interface widget for this may take the form of a gauge, colour bar or arrow in the scene. These widgets may take up valuable view space and clutter the scene with objects that are ancillary to the ‘real’
objects being depicted. They may also require the user to shift their gaze away from the task at hand to observe the measurements.

6.2.2 A solution using force fields

In any 3D virtual environment, a haptic interaction tool can assist the user in the manipulation of objects. With the benefit of haptics (force feedback), the user can reach out, touch and grab objects that they need to work on. This eases the dependence on the use of eyesight to co-locate a virtual tool with the target. With a touchable surface on the virtual objects, a faster, more natural way of working on them is available. Although this is the most common way of using haptics in virtual environments, alternate uses of haptics are possible. Such an alternate is to use auxiliary haptics, i.e. using a haptic effect to assist the user by providing status information about the current virtual environment in the form of a force on their hand. If the program logic needs to pass on advice to the user, as mentioned in section 6.2.1, above, and if this advice has some directional component (i.e. it can be expressed as a vector), then we can introduce a force to the haptic device that effectively guides the user in the correct direction. This force can be applied selectively at certain times and in certain parts of the 3D scene, as required. It is therefore analogous to a force field that has a location, direction and extent, and can ‘wax and wane’ as necessary.

The use of such force fields can have several benefits:

Firstly they can represent advice but not necessarily enforcement with hard barriers such as is described in [118]. We would typically want the user to have ultimate control over the system, since they would be the expert on the particular task at hand – if the computer could perform the task better, there would be no need for a user at all and the task could be completely automated. A gentle force field of variable intensity can advise the user of some rule violation, but not force them to obey the advice. An example might be “you are getting too close to this object”. The user’s reaction might be verbalized as “OK, but this time I still need to go just a little closer before moving away”.

Secondly, a force field can provide continuous advice. It is not binary such as a pop up message. More importantly, the advice can be continuously varying in intensity and direction. This is most appropriate when the information being represented is of a continuous nature.
Thirdly, if used appropriately, it can be very intuitive. In the example of being too close to something, the force is physically pushing the user away, a very intuitive interface; much like an expert’s guiding hand. There is no learning curve or explanation required to understand the mechanism.

Finally, it takes up no view space, does not clutter the visual scene and does not divert the user’s attention away from their task. They don’t need to move their interaction tool to acknowledge it.

6.2.3 Mine Planning Application

To test these ideas we implemented a prototype application to be used by a mine planning consultant when planning a decline into an underground mine (screen shot, figure 6.1). As opposed to a vertical mine shaft, a decline is a steeply sloping underground road that is used by trucks to shift the ore to the surface. The designing task involves drawing the proposed route in a 3D space containing some of the critical geological features. A 3D immersive view is of immediate benefit for producing the initial sketch.

Since the trucks do thousands of journeys over the lifetime of the mine, the goal is to achieve the minimum road distance from ore face to dumping site. However, there are certain restrictions on the route that the decline can take. Obviously it should pass close to, but not necessarily through the valuable ore. As well as this, the safety of the route

Figure 6.1: Mine Planning Application showing sketched line amongst ore bodies. Grey shaft of haptics tool, lower right.
must be taken into account. The road must not be too steep, must have navigable curves and must avoid dangerous areas such as water-logged or unstable rock. These constraints can be implemented as rules within the logic of the drawing program. Such rules fit nicely into the description above, in that they involve three dimensions, are continuous and need to be conveyed to the user without interrupting their work.

We have built an interface which provides the designer with a 3D view of the geology using the CSIRO Haptic Workbench [221] (figure 6.2). This uses stereo and 3D vision, along with co-located haptics via a SensAble Technologies PHANToM [201], to produce a desktop immersive virtual environment. The haptic tool is represented in the scene as a 3D pen, which can draw in space when the tool’s button is pressed (figure 6.2). We have integrated three types of force field into this environment. The first is a repulsive force emanating from dangerous zones within the geology. As the user’s tool approaches these areas, they feel a force pushing them away. This force is inversely proportional to distance, so the closer they are to danger, the greater the force. Users should find this very intuitive and effective in guiding them around the dangerous areas. Often there are dangerous areas in close proximity to each other. In that case the user can ‘squeeze’ the road between two or three zones of repulsion, using their expert knowledge to allow a slight degree of rule violation, if there is no alternative.

The second force field prevents them from drawing a road that is too steep for the trucks to handle. This guides their hand upwards as they approach the maximum gradient. If that maximum is reached, they hit a hard constraint which prevents them from drawing anything steeper.

The third force was designed to prevent the designer from drawing a curve radius that was too tight for the trucks to negotiate. The intention was to detect when a curve was being drawn and provide a ‘haptic groove’ in space that assisted the user to draw a curve in the detected direction and of the optimum radius. We were unable to get a satisfactory result with this part of the

Figure 6.2: The CSIRO Haptic Workbench
prototype. Either the system would detect curves when they were not intended or it would come into effect too late.

6.2.4 Future Work

We are planning to do user trials to compare haptic guidance with various types of graphic and audio guidance. This should be able to determine if using auxiliary haptics can improve the efficiency of a 3D task, and we hope to report on the results soon.

6.2.5 Other possible uses of this technique

The use of haptics to convey rules to a user could be applied in other areas. In CAD programs, haptics could be used to connect object manipulation to some domain-specific rules of the designing task. For example, when designing an automotive engine block, the user might be juggling measurements such as the thickness of the cylinder walls. Typically, there would be an optimum cylinder wall thickness, with a degree of tolerance around it. As the user dragged a slider bar, adjusting the thickness, the haptic force might indicate to them that they were diverging from the optimum thickness.

In surgical training, forces could be used to guide the student in the manner of an expert’s guiding hand. In more abstract domains, such as economics and finance, haptics could perhaps be used to relay varying financial information to the user.

Certain types of data may be better interpreted if they are mapped to a haptic force. An example of this is an application to interpret 3D seismic data, with the auto-correlation being mapped to a haptic force [147]. This produces an effect where features of the data feel like ridges and troughs while other data aspects are rendered graphically. The user can then easily move around the ‘troughs’, perhaps sketching in some annotation.

6.2.6 Summary

The use of haptics can go beyond the representation of the touch and feel of simulated real world objects. It can be used as a user-interface device to convey rules to the user. The interface can be intuitive, continuous and can be accepted by the user as advice without requiring adherence to that advice. It gives the user the ability to override the advice or allow compromise between conflicting constraints.
6.3 Discussion

This paper expands on some of the haptic effects introduced in the preceding chapter, and discusses their use when applied to a mine planning scenario. Section 6.2.4, Future Work, reported on plans to perform user trials on the effectiveness of haptic guidance feedback. These trials were subsequently conducted as part of this thesis and are reported upon in Chapter 7.

The literature reports a growing use of haptic constraints (or virtual fixtures), in assisting a computer user to improve their performance [226] [170]. For example, this technique has been used to assist in a surgical system that incorporates a robot that can cut bone [61]. The surgeon holds the action end of the robot and maintains fine motor control, while the robot applies overall constraints. The surgeon can feel the boundaries of the safe region via a graduated force. However, this does not interfere with the surgeon feeling the bone directly.

6.3.1 Haptic Constraints and Training

A related area involves attempts to use haptic constraints in the form of a ‘guiding hand’ to assist users to learn a dextrous technique. This needs to be differentiated from using haptics to help the user do the technique. The work in this paper concentrates on the latter – assisting the user to do their work, during that work – not assisting them to learn how to do their work. Kuang et al. investigated the use of haptic constraints to provide guidance in training environments [122]. They used a combination of virtual fixtures to guide a user through a complex scene and toward the destination, along a preferred path.

An example of such a virtual fixture was an attracting cone to draw a user to a start of a path. A sphere was used to ‘home in’ on a point and a cylinder helped to attract a user’s instrument to a centre line or alternatively repel them from a centre line. They created a language for assembling these virtual fixtures and demonstrated their use in a situation where a user’s interaction point must be guided through a maze. However, they discovered that there was no improved learning for a haptics group over those given graphics clues only. Other work [75] also showed little improvement in skill-learning by haptic guidance. This result is not surprising when we consider the mechanism of experiential learning of a task involving limb motion. The learning process requires repetition of a sequence, consisting of messages from the nervous system to the muscles of the limb [200]. If a third party, such as a human mentor or a haptic computer program provides forces to the limb, the student is not driving the limb entirely from active
cognitive processes. In the extreme case, the limb could be completely passive and limp, and be driven completely by the outside force. The neurological processes that will be required to perform the same action once the training aid of the external force is removed, are not being exercised and consequently not being learnt. Sacks [192] explained that for learning, sensation must be combined with movement, and together become the antecedent of meaning. The students using a graphic guidance mechanism were able to learn by observing and copying, and therefore actively controlling their motions. In that way, the training environment more closely matched the post-training environment for the task, when all guidance was removed. The haptic trainees were perhaps using the force feedback as a ‘crutch’ which inhibited their learning. This is backed up by Schmidt and Lee [200], who state that when haptics are used to help in a training task, the users can come to rely on the feedback, and therefore not be as efficient when performing the task without the assistance of the feedback.

The work covered by this paper is directed at using the haptic feedback to improve the skills of a worker while they are doing the task, not training them for it. The haptic constraints are intended as an ongoing aid, to advise and improve performance while in use, and are not intended to permanently change a worker’s inherent skill. As an analogy, a ruler helps in drawing a straight line [190], but does not necessarily improve the drawer’s ability to draw straight lines without it. Haptic virtual fixtures are therefore not suitable for training except in the case where they are used to show the student visually what their motions should be, using the student’s own body. In that case, they should be progressively removed to allow the active control of the motions to predominate. This is shown by [69], who found that haptic guidance can only assist learning in the early stages of training, where the student is exploring what is possible.

6.3.2 Force calculations

The paper mentions that object repulsion, line steepness and curvature forces were investigated as guidance aids. The algorithms used to generate these forces are discussed in Chapter 5, section 5.3.1. The paper also briefly mentions the use of forces to provide guiding ‘grooves’ in space for the interpretation of seismic 3D data (section 6.2.5). The method applies equally well to any volumetric data that can be represented as a uniform spatial array of scalar data points. The volume is divided into an array of cells with data points at the corners of each cell. The data points hold a gradient vector representing the direction and magnitude of the derivative of the data with respect to x,
y and z (figure 6.3). The vector field then represents a gradient field in the 3D space. Areas of low (or alternatively, high) scalar value will have surrounding vectors pointing inwards (or outwards). During the haptics loop, as the haptic tool traverses any cell, the force applied to the tool is retrieved from the vectors assigned to that cell, through trilinear interpolation of the eight corner points of the cell. This has been described as a gradient descent method and causes a ‘fall towards’ low (or high) points, which are experienced as grooves or hills and valleys in 3D space[147].

![Figure 6.3 Gradient force vectors (2D representation of 3D grid)](image)

This chapter has proposed the use of haptic forces to assist a user in a dextrous task. However, it does not report on any measurement of user performance with and without those guiding forces. Although it seems logical that a gentle force ‘in the right direction’ should naturally assist a user in any task, this needs to be quantified with user tests. The next stage of work performs such a user study.

---

8 The author was not involved in developing this algorithm, only in adapting it to the haptic workbench environment. The description of the force algorithm is included here for completeness.
Chapter 7.

Haptic Guidance User Study

7.1 Introduction

Chapter 3 described the use of haptic surfaces on objects to promote realism. Chapter 5 and Chapter 6 covered the development of an application which used guiding forces to assist a user in a task. The work so far, however, has made no attempt to measure whether this addition of a haptic capability provides a benefit to the user. This chapter describes a user study designed to do this. The objective of the study was to examine and quantify any benefit that can be provided by haptic feedback in a three dimensional task. Haptic hardware is expensive and the software required to provide the force feedback adds to the complexity of computing systems. A measure of performance increase, if any, would be useful to enable these extra costs to be weighed against any benefit. This study aimed to discover the answers to the following questions:

- Does haptics improve accuracy? If so, by how much?
- Is it a preferred method of providing user feedback over some of the more traditional methods?
- How much does it add to the performance of a user in a complex 3D task?
- Does it constrain the user and therefore limit their freedom?

The study was divided into two experiments. The first experiment aimed to discover if haptic feedback, approximating the mechanical properties of surfaces, could improve a user’s accuracy when performing a task working with those surfaces and to measure the difference in accuracy. The second experiment aimed to determine if using haptic feedback as a form of external guidance can assist a user in a three dimensional task.
This latter experiment was then extended to see how readily the external guidance could be treated as only *advisory guidance* and either accepted or rejected by the user according to their appraisal of its worth. The intention of this was to resolve the issue that, if guidance is provided to a user from a computer, circumstances may occur where the user prefers to ignore the guidance. A computer-controlled system designed for such a circumstance should not overly hinder a user from ignoring the advice. This then gives the user the freedom to apply *their* knowledge and expertise to the problem as well as to receive advice from the computer system, and to reach a course of action that combines the two in the most efficient manner.

### 7.2 Participants

Twenty five participants volunteered to take part in the experiments. Approximately 50% were working in the IT industry, four of whom had virtual reality and haptics experience. The rest were from the general community. They ranged in ages from 13 to 53. All subjects had normal or corrected-to-normal vision, and had no prior knowledge of the experiment. Each subject gave informed consent to participate in the experiments (see Appendix A). Ethics approval for the experiment was obtained from the University of Western Australia.

### 7.3 Apparatus and Setup

The experiments used the CSIRO Haptic Workbench 3D immersive environment [221] described elsewhere in this thesis (sections 3.2.1, 5.2.2), and shown diagrammatically in figure 7.1. The experiment room was partially lit with fluorescent lighting. On a testing table, a 48 cm (19-inch), CRT monitor was supported pointing down onto a mirror. A PHANToM 1.5 device [201] and Magellan space mouse were placed below the mirror as illustrated in Fig 7.1. The software for the experiment was running on a Dell Precision 670 computer with dual 3.2 GHz processors and running the Windows XP operating system. The graphics were produced in 3D active stereo using OpenGL [204] on a NVIDIA Quadro FX 1400 graphics card. The scene was viewed in the mirror using CrystalEyes shutter glasses.
Users viewed 3D scenes through the 3D shutter glasses and worked on virtual models with the haptic device, arranged with a mirror so that the haptic sensation was co-located with the visual location of the objects in the scene.

7.4 Experiment 1: Sculpting

The object of experiment one was to test the hypothesis that haptic feedback assists a user to perform a fine, accurate dextrous task in three dimensions. Participants were asked to carefully sculpt some virtual clay and the experiment measured how accurately they performed, both with and without the haptic ability to touch the surface. Without the haptic feedback participants used their eyesight to determine the shape of the clay and govern the motion of their sculpting hand. With the haptic feedback, they still had this environment but could also feel the surface as they worked.

7.4.1 Scene Description

The virtual 3D scene consisted of a simulated sphere of virtual clay, coloured yellowish-beige, located centrally and with diameter 120 mm\(^9\). The user’s tool was depicted as a pen-shaped handle with a spherical grey sculpting tip of diameter 30 mm. Users held the PHANToM™ in their dominant hand and viewed the virtual tool moving within the

---

\(^9\) In haptic applications real world measurements are used, since users interact with virtual objects with real-world hand and arm motions.
scene to match their hand movements. The real, PHANToM™ arm and the virtual, graphical stylus were co-located and co-oriented as closely as possible during an initial calibration step. In this configuration, user could see their own arm extending beneath the mirror and they viewed the virtual tool at the same location as their (hidden) hand, gripping the PHANToM™, would be. This is a proprioceptive effect which enhances the naturalness of the interface. The user could also place their non-dominant hand on the space-mouse, which allowed them to change the orientation of the clay sphere by simply rolling their hand. When touched with the virtual sculpting tool, the clay surface would deform plastically and the user would feel the reactive force. This sculpting behaviour and the algorithms behind it are described in more detail in Chapter 3.

The clay sphere was modelled, somewhat unrealistically, as a surface skin, rather than a solid volume. This had the side-effect that the tool could be passed through the clay surface (by exceeding a threshold force) to become located on the inside of the clay. This then allowed the user to sculpt outwards, bending the surface out from the interior of the clay. If the user struck the clay surface with a fast action, they could pass through it without causing any deformation at all. The ability to do this was important for the experiment as it then allowed the user to correct indentations by stroking from the inside out, as well as to smooth protrusions from the outside. There was no conservation of volume and deformations occurred only around the vicinity of the tool’s interaction.

7.4.2 Experiment Design.

The experiment was designed to test the null hypothesis; that surface based haptic feedback does not improve accuracy in a dextrous three dimensional task. If this hypothesis can be disproved, we can conclude that haptic feedback improves accuracy in a dextrous task. The experiment employed a two level, within-subjects design with one independent and one dependent variable. The independent variable was the presence or absence of haptic feedback. The dependent variable was the accuracy of sculpting the virtual clay. A significantly lower error rate on accuracy would indicate invalidation of the null hypothesis and provide a measure of the value of the haptic feedback. The level of significance was chosen as 0.05 to match the convention in this area of science.

7.4.3 Method

Subjects were seated comfortably on an adjustable chair in front of the testing table, so that they could easily look into the mirror. Each subject’s body was aligned midline
with the centre of the mirror with hands resting on the table beneath the mirror. Each user was given ten minutes practice on haptic 3D applications unrelated to the experiment, to provide them with familiarity using the hardware and viewing 3D scenes through stereo shutter glasses. They were then given five minutes practice using the experimental sculpting software, to ensure that they understood the necessary techniques.

At the start of the experiment, the virtual clay ball was presented to the user with some prior sculpting, consisting of a series of randomly spaced lumps, already applied to one third of the surface. The user’s task was to smooth out the lumps, returning the clay, as closely as possible, to a sphere. They were given no time limit, and instructed to continue until they felt that they were no longer achieving any improvement. They performed the task twice; once with haptic feedback and once without. Fifty percent of participants were trialled with haptics first and fifty percent with haptics second to remove any learning or fatigue effect.

7.4.4 Measurement
The performance was measured in terms of accuracy of the sculpted surface compared with the intended goal; that of a perfect sphere. The goal was not to achieve a sphere of a specific radius, as this would have been extremely hard for participants to gauge. Rather, the goal was to achieve a surface which was as close as possible to a perfect sphere, regardless of the radius. The sphere was modelled as a triangulated surface, containing 5632 triangles sharing 2818 vertices. The positions of the vertices of the triangles after sculpting were measured as data points. The variation in distance from an averaged distance, in the form of a standard deviation, was used as a measure of error from the perfect sphere. Using the standard deviation compensates for negative and positive errors (depressions inside and projections outside of the perfect sphere). Overall time was not measured, as the intention was to discover the effect of haptics on accuracy during a carefully performed, dextrous task, with no time constraints. Typically, subjects took about 10 minutes to perform each of the tasks.

7.4.5 Results
Twenty four subjects’ results were recorded (one subject was called away before completing). The initial performance-measure was taken as the mean of the standard deviations measured over all participants, firstly with haptic feedback and then without. This was calculated as 0.179cm for the haptic trial and 0.250cm for the trial using no
force feedback. This is shown diagrammatically in figure 7.4. The data recorded appears in Appendix B and is shown in a scatter diagram in figure 7.2.

7.4.6 Analysis

Inspection of the scatter diagram of the results (figure 7.2) indicates that the data are not homogeneously distributed. That is, the residuals of the subjects are widely and non-homogeneously scattered (i.e. not normally distributed around the mean). This then suggests that a T test on the raw data may not be valid in this case. However, if the data are logarithmically transformed (log_{e}[value]), a much more homogeneous distribution is attained (figure 7.3), and this can then validly be used for a paired T test. The means

---

In standard analysis of variance the model requires the residuals from the model are normally and independently distributed with the same variance. Logarithms can be used to stabilize the variance. [214]
of both the raw error data and the logarithmically transformed data are shown in figure 7.4.

The p value returned on the logarithmically transformed error data is 0.0435, which is statistically significant (less than the significance level of 0.05). It can be seen from this result that performance by the subjects in this experiment was significantly better using haptic feedback.

The scatter diagram in figure 7.2 brings to light another interesting characteristic. The largest improvement in skill brought on by haptic surfaces (i.e. the difference between the haptic and non-haptic measurements for each individual) occurs with those participants who had the highest average errors. By calculating the relative improvement of each individual, as a ratio of haptic score divided by an average of their haptic and non-haptic scores, we have a measure of the improvement that haptics provided them relative to their overall ability at the task (equation 7.1). By ordering the scores, it was found that the worst 5 performers had 3.9 times larger relative improvement than the rest.

\[
\text{improvement} = \frac{\text{HapticScore}}{\left(\frac{\text{HapticScore} + \text{NonHapticScore}}{2}\right)}
\]

7.1
7.4.7 Discussion

The null hypothesis, that surface based haptic feedback does not improve accuracy in a dextrous three dimensional task, has therefore been disproved, indicating that this style of haptic feedback does improve performance. The difference in the results has statistical significance. It can be concluded that surface-based haptic feedback assists a user to perform a fine, accurate dextrous task in three dimensions. This presents a compelling argument in favour of providing haptic feedback to tasks where accuracy is important, the task involves 3D manipulation and there is no pressure to work at speed. It says nothing about cases outside these conditions.

The reason for the increased accuracy could be attributed to the user being able to apply a constant pressure against a (virtual) surface as they are controlling their hand motions. This means that their muscles are being controlled in a varying ‘forward’ direction (against the surface). In his book, The Hand, Wilson [248] explains that all complex movement is achieved by the combined action of pairs of muscles aligned in the body to pull against each other. Without the haptic surface to press against, a sculptor needs to control the arm and wrist muscles firstly forward, then backward as they detect that they have penetrated too far. The body uses a different set of muscles for flexing the wrist forward (curling) as opposed to moving it backwards (extending). During forward motion three major forearm muscles work - the *flexor carpi radialis*, the *flexor carpi ulnaris* and the *palmaris longus*. However, backward motion works the *extensor carpi radialis longus*, the *extensor carpi brevis* and the *extensor carpi ulnaris* [143]. This implies that, without surface resistance to provide the backward motion, the muscle action requires two separate mechanisms with the two different sets of muscles; one pressing the hand forward and the other pulling the hand backwards. These two mechanisms must be balanced against each other to achieve a steady movement. It can be surmised that switching between two separate muscular mechanisms is a more difficult control system than constantly varying a single muscle mechanism and therefore it cannot be controlled as accurately. Sheng et al. [207] made use of the same effect by providing a deformable physical prop to press the fingers against during virtual sculpting.

It should be noted that the experiment result applies only to the accurate execution of a task with haptic feedback. It does not address improving the skills of a worker through
training or transfer. The reader is directed to other research [122] [69] [75] [170] [218] addressing this.

The result also indicates that haptic feedback benefits lower skilled users more than those with higher skill levels. This is evidenced by the factor of 3.9 greater relative improvement in the worst performers. It indicates that haptics can be of greater benefit to the performance of novices in a task than those more experienced, a result that could be used in decisions on employing haptics in an application.

7.5 Experiment 2: Force as a User Feedback Mechanism

The object of this experiment was to determine if haptics could provide a useful means of providing feedback to the user about the violation of program rules during a 3D task and if so, to measure its effectiveness compared to more traditional methods. As mentioned in Chapter 5 and Chapter 6, the conventional computer feedback mechanism is that of a pop-up dialog box, often accompanied by an audible alarm. This has some disadvantages in that it is binary (it cannot provide a continuously varying degree of feedback), and it uses screen space, thus competing for visual attention with other tasks the user is trying to perform. In some cases the pop-up message must be acknowledged by clicking an ‘OK’ button and this can be disruptive to the work flow. Other systems involve screen widgets which can provide a degree of continuous feedback, but these have the problem of taking up screen space, possibly amongst a bewildering series of icons, buttons and nested menus crammed with technical language.

A haptic, guiding force may be able to steer a user away from an error condition and not possess these problems. The intention of the experiment was to provide a quantitative measure of the degree to which haptic feedback can assist in this case.

A hypothetical, simplified, mine planning scenario was chosen for the experiment. A mine planner views 3D geological data that is complex and often fills the field of view. Any graphical feedback needs to therefore compete with a wealth of other graphical data being presented. As a mine planner draws a proposed route for an underground decline (tunnel), they need to negotiate several, perhaps competing, constraints (e.g. tunnel gradient, curvature limitations, unstable rock - see section 6.2.3). One of these may be the requirement to avoid zones of danger, such as unstable rock. The experiment was designed to test haptics — as a feedback mechanism, warning of these danger
zones — against the more traditional feedback mechanisms of pop-up messages and screen graphics.

7.5.1 Scene Description

The 3D scene consisted of a graphical model of geological structures created by applying a false colouring scheme to a model derived from geological drill-hole data. The model involved rock structures crossing and folding behind each other. A transparency effect was applied to the different structures to allow each separate rock layer to be visualised, even when it folded behind another. This led to a complex, cluttered, graphical representation (see figure 7.5). Two bright yellow cubes were placed into the scene, one on the left and one on the right. These were the start and end points of the proposed tunnel. The user’s tool was represented as a drawing pen, the position and orientation of which was mapped to the PHANToM™ haptic device. When the virtual pen leaves the starting yellow cube, a red cylinder is drawn along the pen’s path until it enters the end yellow cube.

Eleven randomly placed centres of danger were located within the scene. Each had a ‘strength’ (degree of danger), and a ‘radius’ (extent of danger). The degree of danger receded with distance from the centre in a linear fashion until the radius was reached. Three types of user feedback were programmed into the scene, each being triggered when the user’s actions violated one of the danger zones:

Figure 7.5: Screen shot of mine planning geological scene showing rock layers and tunnel route being drawn.
7.5.2 Experiment Design.

The experiment was designed to test whether there was significant performance difference between the types of user feedback, and especially whether the feedback mechanisms involving haptics were any better than other methods. To show this, the null hypothesis, that haptic feedback methods showed no significant improvement over other methods, would need to be disproved. The experiment employed a 25 x 5 (25 subjects, 5 treatments) two way, within-subjects design. The 5 treatments were the 5 different feedback mechanisms, but 2 of these treatments were combinations of the other 3. The level of significance was chosen as 0.05 to match the convention in this area of science.

7.5.3 Method

Subjects performed this experiment directly after the sculpting accuracy experiment described above (section 7.4), and had the same hardware configuration. The mine planning scenario was read to the participants (see the mine planning experiment plan, Appendix C). It was stressed that they need have no geological knowledge. However, they were asked to imagine that they were experts in the field and the colourful 3D scene they were viewing was meaningful to them. They were given five minutes practice using the system in each of its feedback modes, to ensure that they understood the various combinations of feedback that would be used.

The subjects were required to draw a proposed tunnel from the left hand yellow cube to the right hand yellow cube, keeping the tunnel as short as possible (for mine operations efficiency), but also avoiding areas of danger as much as possible (for mine safety). They performed the drawing exercise five times – each time using a different feedback mechanism to warn them that they were within one of the danger zones. When they strayed into one of these danger zones, a yellow dot would appear at its centre and one of the feedback mechanisms would be activated. The mechanisms were:

a) a pop-up message would appear with an accompanying warning sound

b) a graphical arrow would grow out from the centre of danger towards the drawing point.

c) a haptic force would push the user’s haptic pen away from the centre of the danger.
a) Popup message box, “Danger! Too close”, accompanied by the Microsoft Windows ‘Exclamation’ sound.
b) A red arrow growing from the danger centre towards the user’s current pen position.
c) A repulsive force pushing the user’s pen away from the centre of the danger.
d) A combination of a) and b).
e) A combination of a), b), and c).
A Latin Squares system of randomisation was used to determine the ordering of each of the feedback mechanisms for each of the subjects. The aim of this was to ensure that there was an equal distribution of mechanisms with respect to order of use, as well as with respect to each mechanism’s successor and predecessor. The particular Latin Square used for this experiment is shown in table 7.2.

<table>
<thead>
<tr>
<th>Order</th>
<th>First</th>
<th>Second</th>
<th>Third</th>
<th>Fourth</th>
<th>Fifth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>popup</td>
<td>pop+arrows</td>
<td>arrows</td>
<td>forces</td>
<td>forces+</td>
</tr>
<tr>
<td>2</td>
<td>pop+arrows</td>
<td>arrows</td>
<td>force+</td>
<td>popup</td>
<td>forces</td>
</tr>
<tr>
<td>3</td>
<td>arrows</td>
<td>popup</td>
<td>forces</td>
<td>force+</td>
<td>pop+arrows</td>
</tr>
<tr>
<td>4</td>
<td>forces</td>
<td>force+</td>
<td>pop+arrows</td>
<td>arrows</td>
<td>popup</td>
</tr>
<tr>
<td>5</td>
<td>force+</td>
<td>forces</td>
<td>popup</td>
<td>pop+arrows</td>
<td>arrows</td>
</tr>
</tbody>
</table>

The experimenter manually controlled the choice of feedback from a computer keyboard. Each time a new feedback method was chosen, the danger points were redistributed to a new set of positions. This was to avoid any bias to the results through user learning as the experiment progressed.

7.5.4 Measurement
The experimental software recorded three performance measures:

a) the excess length of each drawn tunnel (over and above the minimum possible - a straight line)
b) the time taken to draw the tunnel
c) the degree of encroachment into the danger zones.

The drawn tunnel consisted of a series of very short straight line segments. When the drawing pen was located within a danger zone, it triggered an algorithm that calculated the instantaneous degree of danger (see equation 7.2).
danger = \left( 1 - \frac{|\vec{c} - \vec{p}|}{r} \right) \times m
\tag{7.2}

where \(\vec{c}\) = the centre of the danger zone
\(\vec{p}\) = the current virtual pen position
\(r\) = the radius of the danger zone
\(m\) = the magnitude (strength) of the danger

The degree of danger determined the strength of repulsive force used for the haptic feedback mechanism as well as the length of the arrow used in the graphic feedback mechanism. It was integrated along the length of each segment within the zone. This danger by distance product was then totalled for all segments from all danger zones to give a total ‘danger encroachment’ score for each drawn tunnel (measure c, above).

7.5.5 Results

Excess Tunnel distance: The results for excess tunnel distance for all the experimental subjects are shown in Appendix D, table D1. The tunnel distance was recorded automatically by the system as soon as the user had reached the end target position for each feedback method. The straight line distance was then subtracted from this to provide the distance over-and-above the minimum possible, to have a zero-based variable.

Tunnel drawing time: The results for tunnel drawing time are shown in Appendix D, table D2. The time was recorded as soon as the system detected that the user’s drawing pen left the starting location and as soon as it entered the end target.

Danger encroachment: This measurement was recorded according to the algorithm described in section 7.5.4. The results for this measurement are shown in Appendix D, table D3.

7.5.6 Analysis

The data collected comprised excess length, danger and time. There were 25 subjects, and each subject performed the task with the five different types of user feedback, which are called ‘treatments’ in the analyses. In addition, these tasks were performed in five different orders in a Latin square arrangement (see table 7.2), with each particular order used for five subjects.

The results can be analysed using a two way analysis of variance, using 25 subjects and five treatments (popup, popup+arrows, arrows, forces, forces+popups+arrows). It uses
the same underlying principles as a ‘T’ test but uses a more complex model accommodating more that two treatments as well as order effects (i.e. effects caused by the order in which a subject experiences the treatments). For each variable, this structure allowed estimation of the differences between treatments, between orders and between subjects in a simple analysis of variance. The treatment-by-order interaction, i.e. whether differences between treatments varied according to the order in which the treatments were given, had to be adjusted for subject differences within the analysis.\textsuperscript{11}

The treatments, subjects and order were all orthogonal (had no effect on each other), however the treatment-by-order interaction was not orthogonal to subject.

The analyses of variance were performed on three variables: excess distance, danger and time, in three different ways to obtain the correct terms for treatment, order, subject and treatment-by-order. However the three analyses for each variable can be combined into single analyses for interpretation.

**Excess tunnel distance:** This variable had to be log-transformed ($\log_e[100 \times \text{excess distance}]$) so that the homogeneity of variation assumption for analysis of variance was satisfied. The residual plots from the analyses of untransformed and log transformed distance are shown in Appendix E. The combined analysis of variance is provided in Appendix F, table F1. Since $P < .001$, there were highly significant differences between subjects and highly significant differences between treatments. Differences between order were not significant at the 5% level ($P>0.05$) and there was no order-by-treatment interaction. The resulting treatment means of the log-transformed data are shown in table 7.6 and shown graphically along with the raw means in figure 7.6.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Mean</th>
<th>Mean of log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup</td>
<td>0.255</td>
<td>2.959</td>
</tr>
<tr>
<td>Popup + arrows</td>
<td>0.202</td>
<td>2.771</td>
</tr>
<tr>
<td>Force</td>
<td>0.153</td>
<td>2.582</td>
</tr>
<tr>
<td>Arrows</td>
<td>0.141</td>
<td>2.484</td>
</tr>
<tr>
<td>Arrows + popup + force</td>
<td>0.170</td>
<td>2.715</td>
</tr>
<tr>
<td>LSD (least significant difference)</td>
<td>0.218</td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{11} The 4 degrees of freedom (4 d.f.) between the five subject groups with the same treatment order appears in both the subject main effect (24 d.f.) and the order x treatment interaction (16 d.f.). No matter which order the terms subject and order x treatment are fitted in the analysis of variance, this 4 degrees of freedom has been counted twice, leaving the residual degrees of freedom 4 less than it should. Consequently 4 degrees of freedom are added to the residual.
This shows that the ‘arrows’ treatment produced a significantly shorter tunnel distance than all other treatments except ‘force’. ‘Force’ produced a significantly shorter tunnel distance than ‘popup’ at the 5% level. Surprisingly, the combination of all three feedback methods resulted in longer tunnels than force or arrows on their own.

**Tunnel drawing times:** As with tunnel distance, this variable had to be log-transformed (log$_e$[time]) so that the homogeneity of variation assumption for analysis of variance was satisfied. The residual plots from the analyses of untransformed and log transformed danger are attached in Appendix E. The combined analysis of variance is included in Appendix F, table F2. Since $P < .001$ there were highly significant differences between subjects, between orders and between treatments. There was no order-by-treatment interaction. The resulting treatment means are shown in table 7.8 as well as graphically in figure 7.7.

### Table 7.8 Drawing time means

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Mean</th>
<th>Mean of log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup</td>
<td>25.58</td>
<td>3.069</td>
</tr>
<tr>
<td>Popup + arrows</td>
<td>21.91</td>
<td>2.952</td>
</tr>
<tr>
<td>Force</td>
<td>17.87</td>
<td>2.704</td>
</tr>
<tr>
<td>Arrows</td>
<td>17.91</td>
<td>2.809</td>
</tr>
<tr>
<td>Arrows + popup + force</td>
<td>20.17</td>
<td>2.789</td>
</tr>
<tr>
<td>LSD (least significant difference)</td>
<td></td>
<td>0.163</td>
</tr>
</tbody>
</table>

![Figure 7.6 Means of excess tunnel distance.](image)
This shows that the force method took the shortest time but was not significantly less than arrows. The popup method produced significantly longer drawing times than all other methods. The combination of the three methods was marginally faster than popup+arrows.

The means for the order are shown in table 7.9.

<table>
<thead>
<tr>
<th>Order of treatment</th>
<th>Mean</th>
<th>Mean of log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject’s 1st treatment</td>
<td>25.10</td>
<td>3.051</td>
</tr>
<tr>
<td>Subject’s 2nd treatment</td>
<td>22.65</td>
<td>2.969</td>
</tr>
<tr>
<td>Subject’s 3rd treatment</td>
<td>20.24</td>
<td>2.872</td>
</tr>
<tr>
<td>Subject’s 4th treatment</td>
<td>17.96</td>
<td>2.723</td>
</tr>
<tr>
<td>Subject’s 5th treatment</td>
<td>17.49</td>
<td>2.707</td>
</tr>
</tbody>
</table>

LSD (least significant difference) 0.163

Individual differences between the five orders could be tested, as done for treatments, however that is really unnecessary as the main result is clear (table 7.9). The average time taken for each treatment (across subjects) steadily decreased as the experiment progressed. The magnitude of the decrease was independent of the particular treatment (order-by-treatment not significant), so the average time over all treatments can be presented (table 7.9) to show the trend. This indicated that either a learning effect was occurring, or users became more confident and casual in their actions. It was noticed that some users were becoming quite cavalier in their actions with the force feedback. This is indicated by the widely varying individual time values when using force.

Figure 7.7 Means of tunnel drawing time
Danger encroachment: As with the other variables, this danger encroachment had to be log transformed \((\log_{\text{e}}[100 \times \text{danger}])\) so that the homogeneity of variation assumption for analysis of variance was satisfied. The residual plots from the analyses of untransformed and log transformed danger are attached in Appendix E. The combined analysis of variance (Appendix F, table F3) produced the means shown in table 7.10 and figure 7.8. Since \(P < .001\) there were highly significant differences between subjects and highly significant differences between treatments. Differences between order were not significant and there was no order-by-treatment interaction.

![Figure 7.8 Means of Danger Encroachment](image)

**Table 7.10 Danger encroachment means**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Mean</th>
<th>Mean of log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup</td>
<td>0.335</td>
<td>3.370</td>
</tr>
<tr>
<td>Popup + arrows</td>
<td>0.287</td>
<td>3.003</td>
</tr>
<tr>
<td>Force</td>
<td>0.209</td>
<td>2.751</td>
</tr>
<tr>
<td>Arrows</td>
<td>0.241</td>
<td>3.065</td>
</tr>
<tr>
<td>Arrows + popup + force</td>
<td>0.203</td>
<td>2.754</td>
</tr>
<tr>
<td>LSD (least significant difference)</td>
<td>0.314</td>
<td></td>
</tr>
</tbody>
</table>

It can be seen that the force treatment performed the best along with the combination of three methods. Both treatments that included force performed significantly better than those without. Popup + arrows performed better than popup alone and the arrows treatment was marginally better than the popup.
7.5.7 Discussion

These results show that using force alone as a user feedback method can improve the user’s skill in a dextrous navigation task in 3D, as is evidenced by the performance in avoiding danger zones. Combining the force with other feedback methods (arrows and popups) did not significantly improve the navigation skill over force alone, and is therefore an unnecessary complication. Tunnel distances were best with the graphical feedback (arrows), but in a real world situation this needs to be balanced against the trade-off of an increased danger encroachment. Users were able to draw more directly to the target but cut through areas that should have been avoided, indicating that the arrows were not serving their purpose as well as forces. Popup messages performed badly on all the three measures, distance, danger and time and should be avoided in an application of this sort.

7.6 Ability to Ignore Haptic Advice

As mentioned in sections 5.3 and 7.1, there may be valid reasons for a skilled user to want to ignore the advice provided by a computer system. It could be argued that a force feedback mechanism might limit this ability to ignore the advice and therefore be less effective as an aid in the overall performance of a task. It is therefore important to achieve a suitable level of haptic assistance without constraining the user too much [242].

The second stage of the experiment was designed to test and measure the ability to ignore the different styles of user feedback.

7.6.1 Scene Description

The 3D scene for this experiment was identical to that used in the previous experiment (section 7.5.1), except that the right-most danger zone was modified to be treated differently. The only visible difference was that this danger zone, located closest to the target end point, had a cube-shaped centre-point instead of spherical, and was coloured red instead of yellow. This danger zone is referred to as the ‘special’ danger zone in subsequent sections.

7.6.2 Method

Participants were instructed to ignore any advice they were receiving from the system when the cubic centre point was showing, i.e. when that advice was relating to the
special danger zone. They should proceed drawing the tunnel path, as if no danger was being indicated. To justify this, they were instructed to assume that they were experts at the task, and for this instance only, could decide that this particular danger-zone advice was in error, and there was no danger.

7.6.3 Experiment Design

The experiment was designed to compare the feedback styles for their influence on the user’s performance when the user was trying to ignore the advice. If any one type of feedback provides a significantly greater deflection in the users’ paths, it can be concluded that it would be less suitable in situations where users are assumed to be inputting their own knowledge and skill into the task. The experiment is especially intended to measure the amount of deflection that the haptic feedback causes, as it was expected that this would be the feedback style that would be hardest to ignore, as it provides a physical force on the user’s hand.

As with the earlier experiments, it employed a 25 x 5 (25 subjects, 5 treatments) two way, within-subjects design. The 5 treatments were the 5 different feedback styles, but 2 of these treatments were combinations of the other 3. The level of significance was chosen as 0.05, to match the convention in this area of science.
7.6.4 Measurement

The system recorded the distance and time as before, but only recorded the danger encroachment for the *valid* danger zones, omitting the special one. Additionally, it recorded the point at which the drawn tunnel entered the special danger zone and the point at which the path left it. When the task was completed, the drawn distance inside the special danger zone was integrated and recorded, along with the direct distance between entry point and exit point. A comparison of these two distances would be a measure of how much the user was deflected from a straight line through the danger, as a result of the advice they were being given by the various user feedback techniques.

![Diagram showing measurement of unintended path deflection due to user feedback](image)

**Figure 7.9** Measurement of unintended path deflection due to user feedback

(see figure 7.9). To perform this comparison, the algorithm in equation 7.3 was used to produce a measurement between 0 and 1, the lower end of this range indicating a better ability to ignore the advice. It could be interpreted as the proportion of wasted distance brought about by the feedback advice.

\[
extraLength = \frac{\text{inside} - \text{direct}}{\text{inside}} \\
0 \leq extraLength \leq 1
\]

where *inside* = distance drawn inside the special danger zone

*direct* = direct distance between entry and exit points of the special danger zone

(7.3)

*extraLength* is a measure of inability to ignore the feedback advice. A low extraLength score indicates that the feedback method is not greatly influencing the user, an outcome that is desired in this case.
7.6.5 Results

The raw results are included in Appendix G. Lower values indicate a better ability to ignore the user feedback. Blank values indicate instances where the participant did not enter the special danger zone at all, perhaps because their chosen path avoided it altogether. These instances were treated in the analysis as ‘missing values’.

7.6.6 Analysis

The analysis of variance was performed on the *extraLength* variable, which had values ranging from 0.0015 to 0.947. The appropriate transformation this variable is the logit [10] transformation:

\[
\log_e \left[ \frac{\text{extraLength}}{1 - \text{extraLength}} \right]
\]

As with the analysis in section 7.5, following this transformation, the homogeneity of variation assumption for analysis of variance was satisfied.

**Table 7.12 Means for Ability to Ignore**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Mean</th>
<th>Mean of logit transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup</td>
<td>0.310</td>
<td>-1.434</td>
</tr>
<tr>
<td>Popup + arrows</td>
<td>0.303</td>
<td>-1.390</td>
</tr>
<tr>
<td>Force</td>
<td>0.192</td>
<td>-2.303</td>
</tr>
<tr>
<td>Arrows</td>
<td>0.283</td>
<td>-1.625</td>
</tr>
<tr>
<td>Arrows + popup + force</td>
<td>0.242</td>
<td>-1.784</td>
</tr>
<tr>
<td>LSD (least significant difference)</td>
<td>0.550</td>
<td></td>
</tr>
</tbody>
</table>

The analyses of variance were performed on the variable in three different ways to obtain the correct terms for treatment, order, subject and treatment-by-order.

However the three analyses can be combined into a single analysis for presentation, which is given in Appendix G. The analysis produced the means shown in table 7.12 and figure 7.10.
The means of logit transformed values are negative because the transformation produces negative values for proportions less than 0.5, which applies for most of the observations.

The force feedback produced significantly lower (more negative) extraLength values (at 5%) than the other methods, while the other methods’ differences were not significant.

7.6.7 Discussion

This is a surprising result. A result which would be evidence in favour of using haptics for guidance would have been that it was equal to, or at least not significantly worse than other feedback methods in this measure. The results of this experiment show that it is in fact significantly better than the other methods. The previous two experiments have shown that haptics can improve a user’s skill when they are willing to comply with the haptic advice, but if they wish to ignore the advice, it seems reasonable to expect that a physical force would be harder to resist and ignore than a graphical prompt. The fact that the force produced better results indicates that users were able to ignore a gentle push more easily than a sudden popup message or a graphical arrow.

One possible explanation for this unexpected result could be associated with the cognitive load on the user at the time. They were drawing a tunnel, trying to obey the computer-generated advice until they reached a point where they needed to recognise that they had entered the special danger zone. They were then confronted by a new

![Figure 7.10 Ability-to-ignore means](image)
condition, requiring them to reverse their behaviour and ignore the advice. Perhaps a gently increasing force produces a lighter cognitive load on an already-occupied user, than a graphical on-screen indicator. An indication of the cognitive load at that point of the experiment is provided by some participants’ very high raw values of the extraLength ratio (close to 1) in the experiment. This indicates that when some users entered the special danger zone and observed the warning, they changed their drawing direction completely around, exiting the zone at a point close to their entry point. This implies a certain amount of confusion interpreting the action to be taken. These occurrences were less frequent with the force feedback.

We can conclude from the experiments in this chapter that haptic effects of different kinds can assist a single user in a 3D dextrous task. The next stage of my PhD work looks at applying those techniques to cases where two users want to work together over a network.
Chapter 8.

Trans-World Haptics

8.1 Introduction

This paper was presented in the “Sketches and Applications” section of SIGGRAPH, held in San Diego, USA in July 2003 and appeared in the ACM SIGGRAPH digital proceedings for that year. It describes work which adapts the haptic virtual environment (discussed previously in the thesis), to a networked situation, allowing users to share the virtual objects from different geographical locations, as if they were both immersed in the same virtual environment. The Sketches and Applications section of the proceedings had a page limit of two pages. For this reason the paper should be read in the manner of a very skeletal overview of the technology involved. Chapter 9 covers the topic in more detail and Chapter 10 covers its application to surgical training.

The paper was the first international report on the pseudo physics algorithms and haptics networking technology that I developed over the previous two years. The co-authors contributed in the creation of the 3D geometries, some of the user interface features and some of the program control. My contribution was in the networking, physics and haptics.

SIGGRAPH attracts delegates with highly technical expertise and interests in graphics, virtual reality, simulation, media and animation.
8.2 Publication 5: Trans-World Haptic Collaboration

Chris Gunn, Matthew Hutchins, Matt Adcock, Rhys Hawkins, CSIRO, Australia
E-mail: {chris.gunn, matthew.hutchins, matt.adcock}@csiro.au

Abstract

This sketch describes a collaborative virtual environment application involving haptic interaction over long Internet distances. We have developed algorithms to accommodate significant latency for certain applications, notably in the medical domain. The results have shown that we can manipulate simulated human body organs, as well as guide each other’s ‘hands’ (and shake hands!) over 22,000 km.

8.2.1 Haptics in Collaboration

In a graphics-only system, a user typically reacts to a change only after it has been consciously processed in the brain, creating, effectively, an adaptive low pass filter. With haptics, the effects of latency are significant, due to the direct exchange of energy between the user and the system, via the user’s flesh, muscles and tendons. These responses can produce instability when there are dynamic objects in the scene and there is significant latency.

It has previously been thought [145] that latencies of greater than 60 ms prevent usable collaborative haptics. [109] found that with solid objects, a 90 ms round trip latency produced stability issues. However, our work has shown that with specialized physics, in surgical environments with soft objects, round trip latencies of 320 milliseconds can be accommodated. This allows haptic environments to be shared by points on the opposite sides of the globe, and may even permit satellite communications to be used.

Plausible Physics

We achieved this by sending all participating forces, including collision impulses, to a single physics ‘engine’ running a ‘pseudo’ or ‘plausible’ physics model. The physics model bears little relationship to reality. However, for our task, the outcome is similar enough to reality to satisfy our needs.
Experimental Environment

The CSIRO Haptic Workbench uses a SensAble Technologies’ PHANToM and 3D stereo shutter glasses. We use a mirror arrangement (figure 8.1) to co-locate the user’s visual and haptic experience, without the haptic equipment or user’s arm occluding objects in the scene [221].

![CSIRO Haptic Workbench](image)

**Figure 8.1: CSIRO Haptic Workbench**

8.2.2 Surgical Simulation

As a test bed for collaborative haptics, we are using surgical training — specifically a cholecystectomy (gall bladder removal). An instructor and student, communicating via a headset/microphone pair, can view, touch and interact with a 3D scene of body organs. The instructor can ‘grasp’ the student’s tool to haptically guide it while the student feels

![Screen shot of surgical training application](image)

**Figure 8.2: Screen shot of surgical training application**
the force of the instructor’s guiding hand. Similarly, the instructor feels any resistance caused by the student. They can also collaboratively push, stretch and pull the organs around the scene, with attached body organs stretching and moving accordingly (figure 8.2). This system also allows the collaborative viewing and annotation of a video of a real operation and a medical scan. The tool can also leave a ‘vapour trail’ of fading tool images following the tool motion.

Software

The system is a toolkit extension to the Reachin API [180], encapsulating the multi-threading and communications code within two new nodes, which can be simply added to a scene-graph. The nodes create ‘remote routes’ between corresponding scene objects, allowing changes to be transmitted using either TCP or UDP at rates up to 1000 Hz.

Conclusion

This technology has the potential to allow surgeons in remote or regional areas to connect to experts in any part of the world, for a mentoring session on a particular procedure. The toolkit nature of this system allows this technology to be easily integrated into a range of other application areas.

=================================================================
8.3 Discussion

This paper was severely shortened by the page limit requirements of the Sketches and Applications section of SIGGRAPH. Consequently, it provides only the briefest overview of the work that I undertook in the development of the algorithms and software to allow networked haptic collaboration. The paper mentions the issues of conscious reaction as well as involuntary reaction to haptic stimuli, and the consequence of these when two users are connected by a latency-prone network. This issue is explored in depth in Chapter 9. Other issues associated with the user interface are discussed in Chapter 10.

Along with the haptic features, the paper mentions the concept of a ‘vapour trail’ user interface feature, which was introduced to provide a trace of the movement of a user’s tool within the 3D scene. The intention was to allow an instructor to demonstrate a complex, dextrous movement by performing it, with a gradually fading trail of tool images, taken as snapshots during the movement as an indication of the historical trace of the action. Our subsequent demonstrations to medical professionals revealed that, although they thought this idea was interesting and novel, they were unsure how it could be used effectively in the training of surgeons. Traditional surgical training methods include a trainee observing an expert surgeon during a surgical procedure and at some later time attempting the surgery themselves. It could be an area of further research to discover if hand and instrument motions are independently remembered by the trainees and whether the display of these motions via other means is beneficial.

The paper reports success at manipulating simulated body organs with latencies up to 320 ms. Elsewhere in this thesis more conservative upper limits on latency are reported. The reason for this discrepancy is that the complexity of the model being manipulated has a bearing on the stability of the system. The model used for the early trials, reported in this paper, was considerably simpler than those described in the following chapters. The most significant contribution to the latency-resilience of the system is the number of linked segments in the duct system attached to the gall bladder. In human anatomy, the gall bladder is connected to the liver and duodenum by a system of branching ducts [81]. As the nearby organs are manipulated by a surgeon, tension is transferred linearly along these ducts, affecting the position and orientation of connecting duct segments, and ultimately the position and orientation of the gall bladder itself. The gall bladder is
also attached to the liver with a webbing of tissue which plays a contributing role in its motion. These ducts and tissue are flexible and have some elasticity. The initial trials of the networked haptic system used a very simple model for this organic interconnection. The simulation of the gall bladder’s motion was achieved by ‘tying’ it to a default position with a simulated spring and then, similarly, tying the connected duct system to the gallbladder. The ducts consisted of very few segments and there was no simulation of the connecting webbing of tissue. Later implementations, reported in the following chapters, used a more complex duct model, with more segments, including the effects of connecting tissue and allowing tension in the ducts to contribute to the motion of the gall bladder. The added complexity required more data to be streamed across the network and vastly increased the number of degrees of freedom of the inter-connected system. As a result, maximum stable latencies were reduced, but are still greater than those reported elsewhere [96] [18] [145].

Te’eni et al. [228] describe collaborative environments as having three commonalities: a shared workspace, WYSIWIS (What You See Is What I See), and a private space that can be uploaded to a public space. This is too limited a requirement, as it focuses too closely on the more established concept of collaboration – that of sharing files, messages or information. They use the examples of shared drawings, pictures, text and chat. The environments addressed in this paper focuses on a different style of collaboration that Te’eni didn’t consider; one where the aim is to provide a sense of presence within a virtual environment. In this case, there should be no concept of a private space and a public space, and no logical step of uploading information for others to share. Te’eni’s ‘shared workspace’ is definitely required, but it is the whole workspace in which the collaborators are immersed. The private workspace does not exist, as it would destroy the illusion of presence. However, the need for WYSIWIS is even more necessary, as the intermediary technology becomes less perceptible, and the participants are less aware that a feeling of presence, together in the virtual environment, is in fact an illusion.

This paper provided an overview of the collaborative haptic application that was developed. The next chapter looks in depth at the experiments that were conducted during development and the algorithms that resulted from them.
Latency and Stability

9.1 Introduction

This paper, **Combating Latency in Haptic Collaborative Virtual Environments**, appeared in the journal, “Presence: Teleoperators and Virtual Environments”, Volume 14, Number 13, June 2005.

The paper discusses in depth the experiments required to form a basis for the result reported in the preceding paper, Trans-World Haptic Collaboration, Chapter 8. While the preceding paper should be treated as an introduction to the work, this paper covers the issues in more detail, explaining the causes of instability across a network and the approach that was used to overcome this in the case of surgical simulation.

I wrote the paper with editing suggestions from the co-authors. The work covered is mine except for that described in section 9.2.11, Surgical Training, which was a team effort involving both myself and the co-authors.

The Presence journal targets a highly technical audience fully conversant with the issues and problems associated with virtual environments and simulation. Some of the earliest haptic applications were developed to overcome problems in teleoperation, and, more recently there has been an increased appearance of haptic techniques in virtual reality applications. This is evidenced by the number of articles in this journal dealing with both haptics and haptic collaboration.

The publishers requested that the image of the simulated surgical environment (figure 9.13, below) be made available for use on the cover of the journal. They also included a brief biography of the authors on the back cover.
Abstract

Haptic (force) feedback is increasingly being used in surgical training simulators. The addition of ‘touch’ is important extra information that can add another dimension to the realism of the experience. Progress in networking these systems together over long distances has been held back, principally because the latency of the network can induce severe instability in any dynamic objects in the scene. This paper describes techniques allowing long distance sharing of haptic-enabled, dynamic scenes. At the CSIRO Virtual Environments Laboratory, we have successfully used this system to connect a prototype of a surgical simulation application between participants on opposite sides of the world in Sweden and Australia, over a standard Internet connection spanning three continents and two oceans. The users were able to simultaneously manipulate pliable objects in a shared workspace, as well as guide each other’s ‘hands’ (and shake hands!) over 22,000 km (13,620 miles) of Internet connection. The main obstacle to overcome was the latency-induced instability in the system, caused by the delays and jitter inherent in the network. Our system involved a combination of an event collection mechanism, a network event forwarding mechanism and a ‘pseudo physics’ mechanism. We found that the resulting behaviour of the interconnected body organs, under simultaneous user manipulation, was sufficiently convincing to be considered for training surgical procedures.

9.2.1 Introduction

In a typical haptic-enabled virtual environment, users can touch both static and dynamic objects. The static objects behave as if fixed in space and are not moved or deformed by the user’s actions. The user feels, through the haptic tool, the surface properties of the object, such as stiffness, roughness and friction. When they touch a dynamic object, they feel these same surface properties, but also are able to move the object in space. The response of the object conveys extra information to the user, about the object’s mass and mass distribution as well as its connectivity with neighbouring objects. This information is often not available from the graphical representation or even the haptic
feel of the object. In a training system it may be important for virtual dynamic objects to behave as closely as possible to their real-world equivalents. Typically, dynamic objects in a virtual environment are modelled to behave according to Newton’s laws of motion\textsuperscript{12}. The amount of force supplied to the object by the user, through the haptic tool, determines the motion of the object that they touch. The dynamic object accelerates while under the influence of all forces acting on it, then continues at a fixed velocity if all forces are removed or are in equilibrium.

Recently there has been interest in the networking of haptic virtual environments. If the shared environment includes dynamic objects, it has been found that severe instability can result if the network has any significant latency [145]. This instability can cause a shared dynamic object to oscillate between two or more points. In surgical simulation, it is necessary to model body organs as dynamic objects attached to each other with elastic membranes, such as vessels and body tissue. Such a system’s behaviour under a force input can be many times more complex than that of a single dynamic object. Networking such a system can lead to violent oscillations, often crazily gyrating, until parts disappear altogether from the visible scene. I discuss some of the reasons for this in Section 9.2.2.

Some research into collaborative haptics [15] [162] [160] avoided this problem by running two virtual environments attached to the same machine. This allowed the study of the human factors associated with a shared touchable scene, without having to grapple with the latency issue. Other work [145] found that latency of up to 60 ms was acceptable for some tasks. This would limit haptic collaboration to maximum distances of about 4,000 km over the standard Internet.

We have found that, for the special case of surgical simulation, the acceptable latency can be extended to approximately 170 ms each way, by using a specialized ‘pseudo’ physics scheme, which, although departing significantly from the laws of motion, is sufficiently close to reality when dealing with body organs. This makes it possible for

\textsuperscript{12} I Every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied to it.

II The relationship between an object’s mass m, its acceleration a, and the applied force F is $F = ma$.

III For every action there is an equal and opposite reaction.
collaborative procedural training of surgical students to occur over distances between any two points on earth. I discuss this pseudo physics in Section 9.2.8.

A typical model of internal body structure can involve a number of body organs that can be independently deformed and moved by the user. Each object may, in turn, move some of its connected or neighbouring objects. There is, therefore, the possibility of a large number of separate events (forces, translations, rotations) that can be occurring at any point in time. To maintain consistency, these events all need to be transmitted across the network to a connected system. It is, therefore, necessary to collect all relevant events and channel them to a networking component for timely dispatch. Our system uses a routing mechanism, configurable by the application developer, to collect these events. In section 9.2.9, we detail the method of channelling these events from the model into a network stream and back into local events, such that the receiving end cannot discern whether events originate locally or remotely. The advantage of this is that the event processing system needs no modification to work in a networked environment.

These concepts were developed on the CSIRO Haptic Workbench, which is described in Section 9.2.10. The concepts were wrapped into a prototype for an application that could be used to train surgeons and medical students. I describe some of the features of this prototype in Section 9.2.11 and then discuss the results of these trials in Section 9.2.12.

This application was developed on top of the Reachin Core Technology [180] application-programming interface (API) - formally known as Magma [221]. This platform allows haptics and graphics to be combined into a single scene-graph.

9.2.2 Dynamic Object Instability

A haptic device, such as SensAble Technologies PHANToM [201] [144] provides a direct exchange of energy with the user holding the device. There is a feedback loop involving the user’s hand, arm, muscles and voluntary and involuntary reactions as well as the robotic force feedback device, network and software involved in the user interface (figure 9.1). Robotic systems with a feedback component can become difficult to control when there is a delay (i.e. latency) in the feedback loop [53]. Fortunately, in stand-alone (i.e. non collaborative systems), the latency is usually so small that the effect is controllable.
In a haptic-enabled, Collaborative Virtual Environment (HCVE), the latency can be much greater, and its effects can therefore be much more significant. The feedback loop involves the physiology of each human holding the haptic device at each end of the network, as well as the network itself and the hardware. Tissue resilience and muscle response of each user seems to play a part, as does their sub-conscious reaction to a sudden force input on the hand [21]. These seem to combine to create unintended force reflections, before the user has had time to respond with an intended movement. Hence, the system is susceptible to instability when these effects are combined with any significant latency.

9.2.3 Force Reflection Experiment

We decided to test the theory that there are some force reflections that occur on a collaborative haptic system, that are involuntary. An experiment was designed to try to measure the individual components of the response of a single user’s hand, when provided with an impulsive force. The force feedback device was a SensAble Technologies PHANToM 1.5. A software module was implemented that could measure the position of the PHANToM at a rate of 1000 Hz. A separate component provides a brief force of 4 Newtons at an unpredictable time, for 0.06 seconds. The tests were performed on 6 subjects of varying abilities with the haptic equipment.

Test 1: Each user’s finger was held against a large heavy (effectively immovable) object. The PHANToM was lightly held against the finger. The force impulse was applied and the position of the PHANToM pen was recorded over the next 0.14 seconds. The graph in figure 9.2 shows a typical resulting movement, which is caused only by compression of the user’s finger tissue. It can be seen that there is a smooth compression of the tissue as the force increases until it starts to smoothly rebound as the force reduces. The point of maximum compression is at 0.0297 seconds. Over the six participants, this maximum compression time ranged from 0.0266 seconds to 0.0318 seconds.
Test 2: Each user’s forearm was held in place by a heavy object, and the above procedure of impulse and position recording was performed. This time, the hand’s muscle/tendon response was included with the tissue compression response. A typical resulting movement is shown in figure 9.3. It can be seen that there is a more complex response. The reaction of both the hand and the finger tissue contribute to at least two bumps on the ‘uphill’ side of the force curve. These bumps would be sent down a network connection to a collaborating machine as two reflection waves.

Test 3: Each user’s arm is held at the elbow – the normal configuration for using the CSIRO Haptic Workbench, since users typically rest their elbow on the desk. The user does not view any image on the screen, however, and they do not have any aim apart from casually holding the pen. The impulsive force is applied and the resulting graph (figure 9.4.) shows the response of the forearm, fingers and tissue compression. It can be seen that bumps exist on both the uphill and downhill sides of the curve, which would be reflected to a collaborating user.
Test 4: Test 3 is repeated, but this time, each user wears 3D stereo shutter glasses and views a target in space. They are given the aim of holding the PHANToM on the target and told to try to maintain it there to the best of their ability. The impulsive force is applied to the PHANToM and the movement under the opposing influences of the force and the user’s reaction are recorded. Figure 9.5 shows the total reaction of the combination of the user’s:

- hand tissue compression
- finger muscle and joint resistance
- wrist and arm resistance
- subconscious and conscious reaction

Figure 9.5 shows that the components of the user’s responses accumulate to produce a complex series of lumps on the ‘uphill’ side of the graph. These would be reflected back to a remote user as a series of ‘waves’ in the case of a networked system. Each
reflection has the potential to cause a subsequent reaction in the remote user, causing another series of reflected waves to travel in the opposite direction. It is not hard to imagine that if these response waves are delayed in the network, an unstable system may result.

Figure 9.6 shows the 4 tests combined into a single graph for each of the 6 users. It is noticeable that the timing of the finger compression peak correlates quite closely to the first hump in the overall trace. Some of the other peaks also correspond to bumps in the overall line. Note that, for each user, the components cannot be arithmetically combined to build the overall trace. This is because each component (finger, hand, arm, overall) was recorded on a different run of the experiment, and the variability of the runs precludes this. Another noticeable feature is that each user has quite different shaped curves (except for the finger compression). This indicates that users of different physiology and experience can have quite different responses to haptic stimuli. This has an impact on damping systems that need to be tuned to the response of a mechanism and explains the difficulty in tuning a damping system to accommodate different human users directly connected into the feedback circuit, as is the case with networked haptics.
9.2.4 Subconscious Reaction

The fidelity of the experiment in Section 9.2.3 was not sufficient to separate the conscious and sub-conscious responses, but there is empirical evidence that such a sub-conscious reaction does occur. It seemed that a user receiving a regularly repeating force pattern is able to control their sub-conscious reaction to the force. In fact, this

Figure 9.6: Combined responses of all participants
phenomenon proved detrimental to our work: We were testing some communications software by progressively adjusting parameters to try to eliminate a feedback-induced oscillation. The changes seemed to be improving the behaviour, and it was approaching a stable state. However, we then discovered that we had been adjusting the parameters of the wrong file, one that was not actually being used! Instead, it seemed that the user, over repeated attempts at the same task, was learning how to control any sub-conscious reactions, and therefore was able to control the oscillations better. This would be an interesting area for future work which may involve aspects of bio-feedback.

9.2.5 Force reflections in a HCVE

It can be concluded from the experiments in Section 9.2.3, that after a force impulse is delivered to a user from a remote system, there are a series of reflected waves passed back to the remote end. The first reflection is from simple compression and ‘bounce’ in the finger tissue. Subsequent reflections are from a combination of muscle/tendon stretching and rebounding, along with a possible sub-conscious reaction. The final reflection is a result of a conscious reaction by the user, and a cognitive response being applied to the PHANToM tool and henceforth to the far end. This cognitive response may be the only one that a haptic system has in common with a graphics only system. In the latter, a user typically reacts to something they see, via mouse or keyboard input, a process that happens via conscious thought only.

When the force reflections occur at both ends of a HCVE, a complex system of feedback can occur. Each force reflection wave passing to a remote user, can feed into the reactions of that user, causing more reflections to be returned from the far end (figure 9.7.). In a latency-free network (e.g. two users grasping a real world object), these countering forces quickly reach equilibrium. However, with latency in the system, the user’s reaction to a force is received at some time after the cause of that reaction occurred, and the causing condition may well have vanished by the time the reaction is received. When the users are both grasping a virtual object, this can result in a mounting sequence of vibrations and oscillations.

Figure 9.7: Feedback loop between two users of a HCVE
9.2.6 Jitter

Network jitter is another factor that can trigger oscillatory behaviour. If the data is received in bursts, or has significant gaps, it can cause the human at each end to over-react and send the system into chaos. In a wide area network (WAN) there is the possibility of losing packets of data, or data arriving out of order. If the TCP/IP network transfer protocol [220] is used, these issues are resolved by the TCP protocol. However, in doing this, lost packets are resent and this can result in a momentary increase in latency. These fluctuations in latency were found to increase the tendency towards instability in a haptically connected system.

Even in a network that has little or no packet loss, such as a local area network (LAN), there can be an irregular packet arrival rate. The experiments were conducted using TCP/IP to link two users’ haptic tools together in an ‘Internet handshake’ across a LAN. The system was configured with a software ‘magnet’, implemented at each end of the network (figure 9.8.). Each user’s haptic tool was drawn towards the magnet by a force. The position of the magnet was controlled by the position of the remote user’s tool. One user was instructed to smoothly move the haptic tool from left to right, thus leading the remote user in the same motion. It was discovered that,
although the software was sending data at a fixed, steady rate of 30 Hz, the data was being received at an irregular rate. The increment in position was smoothly changing with each data packet received, but the timing of the receipt of those packets was irregular. When the system was modified to use the UDP/IP protocol [220], this irregularity was considerably less, and the behaviour was smoother. The difference in jitter is shown in figure 9.9, which shows the variation in arrival times about a mean, for both TCP and UDP. The reason for this is that the TCP protocol has its own buffering, and so will not always send a packet of data when the software instructs it to. It will also attempt to resend lost packets. The implementation using UDP ignores lost packets and continues as fast as possible, to receive the next available packet. For uses such as hand guiding, the timely receipt of data is more important than ensuring that each data position sent is actually received.

9.2.7 Effect of Latency on Physics

A virtual object grasped by two remote users, separated by a latency-affected network, can start to vibrate and oscillate when triggered by a small force impulse. If the virtual object is, in turn, connected to other virtual objects in a spring-mass mechanism (see figure 9.10), these force reflections are transferred to compressions of the springs, and resultant movement of the attached objects. Such a system can quickly degenerate into chaotic movement, often resulting in objects flying completely out of sight. In a haptic system, this result can not simply render the application unusable, but it can be dangerous to the user, as the PHANToM, under such chaotic input, can reach dangerously high speeds if it slips from the user’s hand.

One way to overcome these undesirable oscillations is to apply an artificial damping (force opposed to direction of movement) to the dynamic objects that are being moved by the haptic tool. Trials found that, because the system state information is sampled, the damping adjustments could not compensate for the factors mentioned in Section 9.2.2. Because state information is only available at sampling points, we must apply a stepped damping force, which changes at each sampling point, whereas in the real world, the forces would be continuously varying as the object’s velocity and position continuously changes. This often made the system less stable. A ‘sine x / x’ filter was also tried, using the most common frequency of oscillation discovered during experiments. This failed to control the system under a wide enough range of conditions. A weighted average of past states was also used to predict a future state, but it was also
unable to control the stability over a range of varying latencies and spring strengths. There was no opportunity to perform more than one iteration of the dynamic calculations during each graphics cycle, as there was no ‘slack time’ available in each graphics loop.

In our system, the graphics refresh rate can also change the behaviour, especially when dynamic objects are involved. This is because, in the Reachin API, the dynamic calculation’s time step is numerically linked to the refresh rate.

For these reasons, while it may be possible to ‘tune’ a system to behave consistently under one set of conditions, as soon as any of those conditions change, a different set of tuning parameters may be needed. A self adjusting system could perhaps solve this problem, but it would itself need to detect the system behaviour before making compensating changes, therefore causing a time lag and consequent risk of new instability effects.

**9.2.8 Overcoming Latency**

To overcome these problems, we went back to the basic requirement of our simulation domain. Cavusuglu et al. [46], state that it is important to have task-based performance goals when dealing with teleoperation and this may also be true of collaborative virtual environments. We needed an engineering solution that could involve compromise if it met our performance goals.

After considerable trials, a system was devised that satisfied our requirements. Two steps proved necessary to achieve this. Creating a single physics ‘engine’ and creating a new model to describe the physics.

![Figure 9.10: Body organs as a spring mass model](image)

**Single Physics Engine**

A component of the processing, necessary to calculate the dynamic behaviour of objects, involves the method of collecting all the forces acting on each object in the scene. A stand-alone (non-collaborating) haptic system collects the impulses of the tool on an object at haptics rate (1000 Hz), and includes this along with any other local
forces, such as spring forces and gravity, to generate the correct dynamic behaviour. In a collaborating system, however, we cannot let each node on the network perform its dynamic calculations independently. This is because more than one user may be colliding with the same object simultaneously – and therefore giving it impulse forces. If each system first calculated its position dependent on its local forces, then transmitted its new position to remote collaborators, there could frequently be a mismatch between the position calculated locally and that received from a remote system. The actual position that gets displayed depends on which data, local or remote, was received immediately before the rendering operation. The resulting effect is that some objects can jitter between two positions.

To solve this, one machine on the HCVE was nominated as the physics server. All physics calculations would be carried out on this machine and the results sent to the other collaborator(s). Being an object-oriented design, each dynamic object has its own, encapsulated physics ‘engine’ where this takes place. The input data needed for these calculations consisted of all forces acting on each object. These include user input forces and inter-object spring forces. (In a more complete system, gravity would also be included.) The physics engine knows all the object-interconnection data, so the only force information it needs from the remote machine is the remote user’s input force on that object. Impulse forces from each collaborating user are sent to the physics server, which combines them with local impulses and environmental forces such as springs. It then calculates the object’s new position (according to the incremental approach detailed below (Pseudo Physics)) and returns this information to the collaborators, who then render it accordingly in their scene.

We have a set of C++ classes called “Slave Dynamics” to encapsulate the algorithm running on the client machine. To provide the physics engine with the necessary impulse forces from remote machines, these impulses are accumulated remotely at haptics rates and sent to the server at a slower rate (30 Hz) using the same communication mechanism (see Section 9.2.9) as all other collaborating data.

This method also helps resolve one of the problematic areas of CVEs – that of multiple simultaneous interactions with objects. Without haptics, it is often necessary to create a system of object ownership involving editing-permission requests and responses [43], [96]. However, if we accumulate all forces on an object, we can more accurately mimic real world behaviour, by simply resolving those forces to a single resultant, and moving
the object accordingly. The object moves in the same way that a real world object would move if several people (and other forces, such as gravity) were to try to move it at once. We no longer need to have an object owned by any one user, and no longer need to pass permission requests and wait for permission to be granted before pushing or pulling on the object. Haptics even allows us to extend this system to object editing operations other than simple movement. If two users can haptically interact with each other (i.e. bump and push), then any tool can be allowed to do certain local editing operations (like changing surface features) at any time. This is because, depending on the tool shape and haptic properties, they can exclude each other from the same point in a natural manner by bumping the other tool [161]. This is an example where the added complexity of haptics pays dividends by simplifying other aspects of the system.

**Pseudo Physics**

The second step was to introduce a new, technically incorrect, model to describe the physical behaviour of the objects under the influence of multiple forces. A virtual environment is typically only an approximation of reality in its graphical representation. Surgical instructors have advised us that an approximation of the physics for this particular scenario is acceptable, at least for procedural training.

Under Newtonian physics, the track of a spring-attached object subjected to a user’s force would first involve acceleration under the unbalanced forces. Then, as its position changes, the springs might stretch creating potential energy that eventually decelerates the body to a point where it may change direction. A diminishing series of variations in position, velocity and acceleration will occur until the body is at rest again. We had previously found that, even with the addition of filtering and prediction algorithms, Newton’s laws of motion could still easily result in instability. Because the prediction and filtering could only happen at discrete sample times, the ‘corrections’ in velocity at each time step would often cause an ‘over-shoot’ in the position at the next time step, thus requiring an even greater correction to be applied. Surprisingly, damping adjustments often amplified the oscillations.

However, the specialized area of surgical simulation has a particular advantage; the 3D objects that are manipulated are typically highly damped. An organ’s oscillations about its equilibrium point are typically so small that we can consider them to be insignificant. Surgical objects are so damped, in fact, that it is not unreasonable for a simulation environment to assume that they always have zero momentum. When this lack of
momentum is incorporated into the calculations, it becomes apparent that the objects also have no acceleration and are instantaneously assigned a velocity proportionate only to the forces acting on them. Once we make this assumption, we can develop a totally different (and technically incorrect) physics of body movement, which can come closer to reality than that achieved by trying to accurately follow the laws of motion on a time sampled computer system. It is, in fact, a short cut to avoid complex simulation of the viscous body fluids that encase the body organs. We needed an object to, fairly slowly, slide from one position to another under the influence of the user’s tool and it’s connected ‘springs’ (tissue membranes). Body organs are unlikely to rebound off surfaces, continue in space at a fixed velocity, follow a trajectory when ‘thrown’ by a surgeon, or oscillate around an equilibrium point until coming to rest. They are much more likely to slide under the influence of any force, and then stop, almost immediately, after the force is removed, or after multiple forces reach an equilibrium point. This is a feature of the surgical environment - it would not apply to objects in many other situations, such as ball sports, space simulation or war games. Because the objects move slowly and have negligible momentum (they do not overshoot their equilibrium point), we can strip out much of the physics calculations necessary to simulate the laws of motion. This, fortunately, removes much of the cause of oscillations in a networked, latency-affected environment.

At each time step, a dynamic object knows its current position, and all forces acting on it (from both users’ tools, and attached spring extensions). It can therefore calculate its predicted equilibrium position. The object then calculates a vector from its current position to the predicted equilibrium point and subdivides this into a number of steps. It then repositions itself incrementally towards the equilibrium position, by one of these steps. No acceleration or momentum is involved - it is effectively an animation towards the equilibrium point. At the next time step, the equilibrium point is likely to have changed, due to the forces changing. Typically, this slightly changes the equilibrium point, and the above algorithm’s incremental approach to that point is redirected marginally. The sequence halts when the distance to the calculated equilibrium point is below a certain threshold. The algorithm is as follows (equation 9.1):
A ‘damping factor’ is needed to be able to modify the ‘feel’ of the object as it is being moved and replaces the modelling of any viscous fluid embedding the organ. It can be loosely associated with the difficulty of moving an object. Objects with a high damping factor feel heavy and sluggish. Low damping factors produce objects that move more freely.

We no longer calculate an object’s acceleration from the resultant force acting on it, and then, from that, its velocity and finally its position. Instead we convert the accumulated forces to a predicted equilibrium point directly, and move incrementally towards this point.

The method can be summarized in equation 9.2:

\[
\text{new\_position} = \text{old\_position} + k \times \text{resultant\_force}
\]

where \(k\) is the pseudo damping factor

This method removes the following steps from a Newtonian physics calculation:

- Calculation of acceleration from applied forces and body mass,
- Calculation of velocity from previous velocity and acceleration,
- Calculation of position from previous position and velocity,
- and replaces it with the single step of:
- Calculation of position from applied forces.

Since our aim was to create a system that could be used for procedural training of surgical students, this pseudo physics solution produced satisfying results. It has been demonstrated at a workshop at the Royal Australian College of Surgeons (RACS 2003 Scientific Conference, Brisbane, Australia). Their comments mainly addressed the
strength of the forces that were assigned to inter object attachments and these can be easily adjusted. The behaviour of the object movement was not questioned. Note that our decision to use this simplified physics was not to save on processing steps (although this was a side benefit) but rather to create a physics model that was less susceptible to instability.

9.2.9 Event Collection

A CVE may require a large number of data elements to be transmitted between collaborators. Bogsanyi and Miller [26] suggest that there may be a possibility of reducing the amount of data transmitted, by using a tool based synchronization system. This involves sending only the tool data (position and orientation) between machines, and letting the remote representation of the tool interact independently with objects on the remote machine. In earlier work in our laboratory, we had found that this method fails when there are dynamic objects involved and there is network latency in the system. The arriving remote tool’s position is always historical, having occurred at some time (the latency) in the past. If it is then allowed to interact with dynamic objects locally at the current time, inconsistent behaviour can occur on two collaborating machines. A consequence can be a dramatic divergence of the two systems’ behaviour from that point forward. It may be possible to avoid this problem by time-stamping all data packets, and keeping a history of movement of dynamic objects. The receiving system can then ‘wind back time’ so that a data packet can act on a matching (historic) scene representation, but, as well as much increased complexity, it may introduce further discrepancies if the local user has already changed the scene more recently.

Another problem with a tool-based system is that it requires all collaborators to have access to similar user interface devices. In an object-based system, the remote collaborators don’t care how the changes were made; they just receive the changes themselves. This is demonstrated by our related implementation of a non-interactive viewer, which can remotely observe the haptic collaboration as it occurs, but has no interaction mechanism (and therefore no tool calculations) itself.

For these reasons, it is necessary to send data describing each dynamic object’s position and orientation across the network, as well as the user forces acting upon those objects as they push or pull on them with a haptic tool. We have a set of ‘SlaveDynamic’ objects, which detect user interaction and forward those forces on to the physics server. They do not move themselves until the server returns their new position and orientation.
We developed our systems on the Reachin API [180], which is a scene-graph API that has C++, VRML and Python interfaces and provides both haptic and graphic rendering of the scene. It contains a useful routing mechanism, which was able to be modified to collect and send events across the network. Based on the VRML scene-graph standard [92] it has a convenient system of routing fields between scene graph nodes. This concept has been extended by us to ‘remote routes’ that exist across a network.

The mechanism of remote routes is used to transmit the minimum amount of data in a timely fashion to a collaborating system. Each field whose value cannot reliably be determined independently is routed to a RemoteField object. These objects are members of a RemoteConnection, which multiplexes the RemoteFields and sends the data across a network TCP/IP connection, using the Netwrap library [70], to a matching RemoteField on another system. The receiving RemoteField is in turn, routed to a corresponding part of the scene-graph, which is updated accordingly (See figure 9.11).

Consider two connected haptics systems, A and B. When an object on system A moves, a sequence of events happen:

- its position field changes,
- its corresponding RemoteField gets a notification via the route,
- the new position is transmitted to system B,
- a RemoteField receives it,
- a local route is triggered and the corresponding position field on system B changes,
• the object on system B moves

The TCP/IP remote routes are suitable for most of the dynamic objects in the scene. However, they fail when used for tightly coupled haptic fields (e.g. hand-to-hand guiding). This is because the data is only received at graphics rates, and travelling via TCP, there can be significant variance in the transmission rate (see Section 9.2.6).

Therefore, we detect the tool position at haptics rates (1000 Hz) and route it to a UDP-connected RemoteField, which can be set to transmit at a steady, pre-set rate. Experimentation has found that 400 Hz is sufficient for smooth behaviour. UDP was chosen because we do not really care if some packets are lost; we simply need to get the next packet as soon as possible. For hand guiding, the remote tool position is connected to the local tool position with a simulated spring, or magnet, enabling each user to feel each other’s movements.

Haptic interfaces need to be refreshed at about 1000 Hz. If the refresh rate drops below this value, undesirable buzzing and vibrations can occur when in contact with simulated surfaces. For this reason, it is necessary to dedicate a separate program thread to handling the haptics, and it is preferable to run the application on a multi-processor machine. The Reachin API provides us with a haptics thread, at 1000 Hz, and graphics thread, typically running at around 30 Hz. Some additional processing can be added to the graphics thread without noticeable penalty. In a CVE, however, it is necessary to create extra threads to handle the communications. In our client-server system, we need up to three extra threads on the server for each client and also a single thread to handle client connection requests. The client similarly needs up to three separate threads for communications. One of these threads handles the reading of the less critical data that can be sent at graphics rates, or alternatively, whenever the source of the data changes. The second and third threads handle the sending and receiving of time-critical data (e.g. data that directly affects the behaviour of the haptic tool). For our purposes, it is sufficient to set these threads to run at 400-500 Hz, but it is also possible for them to run at rates up to 1000 Hz to match the haptics refresh rate.

Our extension to the Reachin API hides the multi-threading details from the application developer. A developer can describe their scene in a VRML-style ASCII file [92], including the communications code, by adding instances of two new VRML node types, RemoteConnection and RemoteField. They complete the networking of their application
by adding the ASCII lines describing the routes from those nodes to their dynamic objects.

9.2.10 Haptic Workbench

The trials and the surgical application were developed for the CSIRO Haptic Workbench (figure 9.12), a platform that has proved suitable for working with 3D models for long periods of time without fatigue. We use a SensAble Technologies’ PHANToM as a 3D input device and stereo shutter glasses to view the model in three dimensions. For training in 3D dexterous tasks, it is preferable for the input location and forces to be co-located with the objects in the graphical scene (Stevenson et al., 1999). This enhances the illusion of reality, as well as more accurately simulating the actual hand-eye co-ordination needed for similar, real-world tasks. This is vital in the development of virtual reality training applications, such as remote surgical training. If, however, we were to simply place the PHANToM in front of a stereo display, it (and the user’s arm and hand) could occlude part of the scene. Worse than that, it could occlude objects logically in front of the PHANToM, as well as behind, effectively destroying the 3D illusion. To solve this, we place a computer screen pointing downwards onto an angled mirror. The user sits comfortably at a desk, looking into the mirror with hands resting under and behind the mirror. The virtual objects are then seen located exactly where the hands and the PHANToM are located. However, instead of the real world equipment, only the virtual world objects are seen (see figure 9.12).

9.2.11 Surgical training

A surgical procedure training system was the test-bed for our collaborative haptic trials. The scenario is that of a cholecystectomy (removal of the gall bladder), with an
instructor and remote student, both working in the same virtual space (see figure 9.13). They both wear a headset/microphone pair and communicate via RAT (Robust Audio Tool). They view a 3D scene comprising liver, stomach, kidneys, gall bladder, heart, colon etc. and both users can see each other’s tools in the scene.

The instructor can grasp the student’s tool, to haptically guide it (along with the student’s hand) to a point in the scene, such as the gall bladder. The student feels the force of the instructor’s guiding hand and the instructor feels any resistance by the student. Similarly, the student can guide the instructor, if necessary. The instructor might grasp one part of the gall bladder and tell the student to grasp another part. Between them, they can move, stretch and pull the gall bladder around the scene. The attached bile duct moves accordingly, stretches and provides resistance. The instructor might then choose to draw arrows, circles or other annotation in the 3D scene, while explaining the stages of the procedure. They can independently zoom and pan their viewpoints, but also, the student can ‘slave’ to the instructor’s view. This allows the instructor to fly them both around the scene.

They can jump to a video player (embedded in the 3D scene) showing a real cholecystectomy. They both have collaborative video control and can also annotate the

![Figure 9.13: Screen shot from surgical training application showing two tools manipulating a ‘gall bladder'](#)
video screen to discuss a certain point. They can also jump to a ‘light table’ where a medical scan can be viewed and similarly annotated, or a white board for collaborative sketching.

If a particular dexterous technique was being demonstrated, the instructor can switch on the ‘vapour trail’ facility. Both instructor and student now see a trail of tool images following the instructor’s tool. These slowly fade with time, but linger long enough for the student to practice, following the action with the local tool.

### 9.2.12 Results

We were able to run the collaborative surgical simulation between two immersive virtual environments running across the Internet between Canberra, Australia and Stockholm, Sweden. Since the connection went via the Pacific, the U.S.A. and the Atlantic, the total distance was about 22,000 km. This produced a latency of around 170 ms, one-way. The behaviour was smooth and stable with both users simultaneously manipulating a soft, elastic ‘gall bladder’. They were also able to both guide each other’s tool in the scene feeling the bi-directional force.

These results could be generalized to any situation where the participants need to move objects but do not need to send them on trajectories. The every-day task of moving things on a surface, such as a table, could well come into this category. In these cases we can ignore momentum and acceleration and simply calculate an object’s position directly from the force and time for which it is applied.

### 9.2.13 Further Work

Related clinical trials are scheduled for the third quarter of 2004.

It is also planned to add clipping and cutting capability to the ducts. We also hope to be able to model, more realistically, the deformation of the human tissue involved – the current model is very simplistic. It is not clear whether, in a CVE, accurate tissue deformation simulation will create a prohibitive amount of data to be transmitted across the network. There is also work planned to test a CVE where a high performance computing resource is accessed via the network to do Finite Element Modelling (FEM) calculations for these deformations and to distribute them back to the collaborating haptic systems.
9.2.14 Conclusion

Until now, there seems to have been some pessimism in the search for usable haptic collaboration over long distances. It has been thought that the unavoidable latency of (even optical) networks has placed a restrictive upper limit on the network length between collaborating haptic systems. These trials have shown, on the other hand, that for some suitable applications, there are ways around these issues. A usable surgical training system, with dynamic objects, allowing users to simultaneously hold, pull and stretch them, is possible from opposite sides of the world. Not only that, they can guide each other haptically around the scene, opening up the possibility to teach dexterous tasks by example, or simply to drag another user’s tool to an interesting spot instead of trying to describe it in words. Having succeeded in their task, they can even ‘shake hands’. It is interesting to note that the latencies accommodated by this system approach the same order of magnitude as that involved in satellite communications, opening up the possibility of using those channels for haptic collaboration.

This case study could be generalized to apply to any area where objects in a collaborative virtual environment need to be moved by two users, but they do not need to be sent on trajectories. Apart from surgical simulation, such examples could include repositioning objects on horizontal surfaces such as tables and desks or within viscous fluids such as under water simulations. In these cases, if the object’s change in position is calculated from the product of force and time alone. \( dx = k \times \text{force} \times dt \), a greater resilience to latency can be achieved. The pseudo physics would not be suitable in the simulation of ball sports, space scenarios or war games, where objects may need to be sent on trajectories. In a collaborative system, forces from all collaborators should be calculated in one algorithm and the resulting movement redistributed to the collaborators, to achieve consistency between collaborators. The system of using a single physics server will probably not scale up to many more than two users, but we have been advised that this should not be a limitation in a surgical training system, as two surgeons working together is the most likely scenario.

========================================
9.3 Discussion

This paper describes issues associated with using haptics across a network connection. The network connection used for these trials was between Canberra, Australia and Stockholm, Sweden using consumer networks. Successful, smooth dual manipulation of deformable simulated body organs was achieved, despite the inherent network latency and jitter (variability in latency). However, further tests showed that the performance of the system could vary considerably depending on the time of day. During Australian daylight hours, there were periods when no traffic could get through, and the system would freeze for several seconds. It is thought that this was caused by large data transfers happening on the network at the same time. It is important to note that the authors are not suggesting that any such system could be used for real remote surgery, only for training for surgery. Any interruption to data flow caused by network irregularities would likely be catastrophic in a real surgical situation.

Further tests were conducted between Canberra and Stanford University, California. These utilised the CeNTIE network [48] within Australia, the Internet2 network [106] within the USA and AARNet [1] using the Southern Cross Cable Network [217] across the Pacific Ocean. This was followed by demonstrations at the Internet2 Member Meeting in Austin, Texas in 2004, where delegates at the meeting were instructed in surgery from my laboratory in Canberra. In these cases there were no problems with network traffic, and in the case of the connection to Texas, the system worked continuously for the 6 hour demonstration session. In May 2004, the system was also used to instruct both a student and class in gall bladder surgery across the Pacific Ocean. This event is described in Chapter 11.

Long distance tests are hard to organise and conduct, due to the time zone differences and work commitments of colleagues at other institutions. To allow for stress-testing the system, I developed a mechanism to induce a simulated latency on a local area network within the lab. This involved creating a message queue of configurable length and redirecting data packets to the queue instead of the network. Packets would then be regularly removed from the front of the queue to be sent to the actual network. The length of the queue, therefore, determined the amount of simulated latency of delivery to the destination. With this in place, it was possible to test the physics model described in the paper and to find the upper limit of latency at which stability could be maintained.
With two users manipulating both the gall bladder and connected duct system, it was possible to maintain stability of the system described with 290 ms latency. However, as mentioned in the paper, jitter is also a contributing factor in inducing instability, and since the latency simulation system had no variability in the delivery of packets, this was not measured in this experiment. However, Gutierrez reported that a haptic hand-guiding application (AutoHandshake) was resilient to packet loss but sensitive to jitter and latency – rapidly becoming unusable beyond a certain point [88]. In my application, with jitter included, it is expected that the stable upper limit of latency would be below the measured 290 ms. Dev et al. [66] found that there was no graceful degradation with increasing latency in a networked haptic surgical simulation and movements rapidly became unstable. Crawhall [57] argues that latencies of greater than 50 ms prevent the use of haptics across a network. He suggests that this limits collaborating haptic applications to be conducted on networks of only about 3000 km. Regardless of jitter, my work has shown that latencies far in excess of this are in fact possible, for certain types of model behaviour.

The Further Work section of this paper, section 9.2.13, mentions the aim of adding clipping and cutting of ducts to the simulation. This work has subsequently been done, and is reported on in Chapter 10. It also mentions the need to model the tissue deformation more accurately. This work has not been done in our laboratory but is being pursued elsewhere [229] [216] [129].

In their study of motion simulation algorithms, Agarwal et al. [5] concluded that “Also needed are algorithms that can let go of the true physics just enough to meet the constraints” (page 558). In my work, the pseudo physics (sometimes referred to as Aristotelian Physics) solved the problem of latency-induced instability. The constraint in this case is the network distance used to gain the collaboration, and the latency caused by the immutable speed of light, as well as queing delays within network routers.

This chapter described the design decisions and algorithms required to achieve a believable physics model that could allow sharing of deformable elastic objects over a network, and the building of a surgical training application around them. The next chapter will describe additional features that I added to this system to enhance its usability as a training environment.
Chapter 10.

Surgical Training

10.1 Introduction

This paper, Using Collaborative Haptics in Remote Surgical Training, was presented at the WorldHaptics 2005 conference, at Pisa, Italy in March 2005. It takes the work briefly described in section 9.2.11 of the preceding chapter, and expands on its features and implementation. It also describes additional features that I added following the publication of the earlier paper.

The conference is recognised as a leading haptics conference, being a biennial union of the EuroHaptics conference and the IEEE Virtual Reality Conference’s Haptics Symposium. The delegates were highly technical and conversant with issues to do with haptics, simulation and virtual reality.

Contribution: I wrote the paper with proof-reading and editing assistance from the co-authors. The software development involved in this advance on the system, including diathermy of tissue, rupture, bleeding and clipping and cutting of ducts was 100% my work. Two of the co-authors were previously involved in development of anatomical models and program control logic for an earlier version of the application before these features were added, as mentioned in Chapter 8 and Chapter 9. The other two co-authors undertook the user questionnaire analysis, described briefly in section 10.2.6 of the paper.
10.2 Publication 7: Using Collaborative Haptics in Remote Surgical Training

Chris Gunn*, Matthew Hutchins*, Duncan Stevenson*, Matt Adcock*
Patricia Youngblood†
(*) CSIRO, ICT Centre, Australia
(†) Stanford University School of Medicine
E-mail: Chris.Gunn@csiro.au

Abstract

We describe the design and trial of a remotely conducted surgical master class, using a haptic virtual environment as an integral part of the learning process. In the trial, we linked a haptic virtual environment in Canberra, Australia with a second installation at Stanford University, California. We were testing several haptic components of the system, and whether collaborative haptics could be useful in teaching surgery at a distance. We were also interested to see if an audience could be engaged in the instruction. The participants used features such as manipulating body organs, diathermy and clipping and cutting of ducts. The audience followed each student’s performance on a large 3D screen while waiting their turn at the interface.

A key aim of the application was to produce a shared sense of presence in the virtual environment. The responses of the audience and participants in this regard were collected and results are presented.

10.2.1 Background

Collaborative (networked) virtual environments (CVEs) have been found to have several benefits when groups or individuals need to share information or skills. Singhal and Zyada [211] list some of these as “a shared sense of space, a shared sense of presence, a shared sense of time, a way to communicate and a way to share data”. It has also been shown [15] [196] that haptic (force) feedback significantly improves perceived virtual presence in a CVE, and that it also enhances performance considerably in a shared, dexterous 3D task. Salilinas, Rassmus-Grohn and Sjostrom [196] found that force feedback improved performance and a sense of presence in a cube manipulation task.

However, this field presents several challenges. These include communication delay (latency) and logical consistency between separate representations of the environment [145] [160]. When haptic feedback is used as an interface mechanism, solving these
issues becomes more critical, due to the direct and immediate user input at each end. Kammermier et al. [112] state that as well as providing information about the environment, haptic interaction also implies a bilateral exchange of energy between the human operator and that environment. This produces a feedback loop that is susceptible to instability.

To use collaborative haptics in a real world application, over a real network, we needed to overcome the stability problem. Kim et al. [115] were able to achieve usable haptic collaboration across the Atlantic Ocean by introducing a predictive algorithm for collision detection, as well as three layers of damping. Our work [87] has shown that in certain circumstances, for example highly damped environments with soft objects, we can use a specialized physics model to withstand latencies of around 200 milliseconds. This allows such haptic environments to be shared by points on the opposite sides of the globe. We cover how this algorithm was applied in this trial in section 10.2.3.

Hespanha et al. [96] discuss methods of resolving the problem of simultaneous access to the same object, by using object ownership and locking. In surgery, we are simulating real world activities and are therefore able to circumvent this complexity by allowing the combination of all applied forces to resolve the conflict as would happen in the real world. This is also covered in section 10.2.3.

Virtual Reality technology is increasingly becoming an important component in modern surgical training. Cosman et al describe the limitations of the current apprenticeship model of training, including limited exposure to a variety of operating procedures and the need for “continual high-quality feedback on performance” [56]. They conclude that “There is no doubt that simulators will play a role in the training of future generations of surgeons”.

At SIGGRAPH 2003 we presented a pseudo-physics model, which can be used to help control the tendency to dynamic instability and allow long distance haptic collaboration [84]. This experiment extends that work in the directions of the quality and usability, and to include wider audience involvement. For this demonstration a temporary broadband internet connection was established between the two venues: a conference auditorium holding a simulation technologies conference (SimTect 2004) in Canberra, Australia and the Stanford University School of Medicine in California in the USA. The class was located at the conference in Canberra and the instructor was located in an office environment at Stanford University.
10.2.2 Hardware

The conference presentation environment used an immersive, hands-in, networked haptic environment, high quality, multi-screen video, echo-free audio and a large scale 3D projection screen. The entire system made use of a high bandwidth connection between the two sites, made possible through three academic and research networks; CeNTIE [48], within Australia, AARNet [1] using the Southern Cross Cable Network [217] for the trans-Pacific connection and Internet2 [106] within the United States.

For the immersive haptic interaction we used two CSIRO Haptic Workbenches [221] (figure 10.1). One was installed on the stage at the conference centre in Canberra and another at Stanford University School of Medicine, where the surgical instructor would be based. The workbench contains a SensAble Technologies’ PHANToM 1.5 [144] [201] as a 3D haptic input device and active stereo shutter glasses [110] to view the model in three dimensions.

The application at each end was running on a Dell dual processor 2.8 GHz PC running Windows 2000, fitted with 3DLabs Wildcat 3 7110 graphics cards, with active stereo output. For the class, the graphics output was also directed to an active-to-passive converter box and then to the two passive stereo projectors for the audience view. The projected 3D scene was displayed on a 3m square reflective screen on stage and the audience were issued with passive stereo glasses.

A camera, miniature screen and microphone were installed within the CSIRO Haptic Workbench, giving users a close-up view of their collaborator. A broadcast quality video conferencing system enabled all participants (including the audience) to converse in a natural way. At the conference hall we installed two large plasma screens, each with a camera and echo cancelling microphone attached, allowing face to face conversations at several points on the stage. A high bandwidth connection allowed the use of Digital Video (DV) over IP transmission, providing broadcast-quality video and audio with no jerkiness or flicker and very little latency.
10.2.3 The application

The surgical training application has the scenario of a cholecystectomy (gall bladder removal), with an instructor and remote student working in the same virtual space. The training system has several views, which can be visited by users individually or in unison. The primary view is of a 3D model of body organs such as liver, stomach and kidneys, which was obtained by segmenting data from CT scans.

The training system has been designed with the philosophy of supporting discussion between the instructor and student about the anatomy and key steps involved in the procedure. It is not intended to be a high fidelity simulation. Instead, the goal was to include indicative representations of objects and actions that can serve as a starting point of teaching discussions and aids to memorization. These indicative representations are augmented with the other views, as discussed in Section 10.2.4.

Haptic guiding hand

Within any view, the instructor can remotely grasp the student’s tool, to haptically guide it (along with the student’s hand) to any point in the scene (figure 10.2). The student can feel the force of the instructor’s guiding hand and the instructor can feel any resistance by the student. This enables the instructor to quickly move a student to the correct position at any time or to help the student apply the correct force to organs such as the cystic duct. The guiding hand is implemented by providing an attractive force on each machine. The position of the attractive force is detected from the tool position on the remote machine at the haptics frame rate (1000 Hz) but we found that it is only necessary to transmit it across the network at 400 Hz. The strength of the force felt by the remote user, towards this point is calculated at the haptics frame rate with a
sinusoidal force function shown in figure 10.3.

The graph shows a dead zone around the co-location point, a means of avoiding vibrations. The force is also smoothed between successive haptic cycles, resulting in the student feeling a gentle pulling force to the instructor’s tool once within range. The attraction is activated with the button of the PHANToM device, and only comes into effect if the button is initially depressed whilst inside the active zone.

**Deformation of body organs**

Most organs in the model are deformable, simultaneously by both instructor and student (figure 10.4), allowing either participant to push, grasp and stretch the body organs. The pliability of each organ is set differently to demonstrate the variability possible, e.g. the liver is configured to be stiffer than the stomach.

Due to processing power limitations, we built all organs using a surface mesh. No volumetric calculations are performed. This concession was not perceived as a problem by the medical professionals who took part in the trial. The force feedback is proportional to the extent of deformation of the point of contact, while the deformation of the rest of the object is shown graphically. On the remote machine the deformed shape needs to be felt haptically, to give the remote user the ability to grasp the shape in its deformed state.

The deformation behaves in three different ways depending on the requirements of each organ. Some organs, such as the liver, are considered to be basically fixed in place but elastic. Some organs are set to be permanently moveable to some degree. These are set to have some plasticity. This means that the longer that they are held in their deformed shape, the closer to that shape they will be when released.

The ability to set an object’s elasticity and plasticity is built into the surface model. For each organ, two sets of surface coordinates are stored: the *original coordinates*, and the *deformed coordinates*. The haptic feedback is determined by how far the haptic tool is moving a point in the *original coordinates*. As well as providing a force to the user, this movement is used to calculate an offset for each of the other coordinates of the surface, depending on stiffness parameters and distance from the contact. The *deformed coordinates* are each placed at these offsets from the *original coordinates* and are used for rendering the shape graphically. If an organ has some plasticity, the *original coordinates* are incrementally repositioned towards the *deformed coordinates* over time.
The third deformation type is needed to simulate the gall bladder and cystic duct system. The gall bladder is connected to ducts which branch and join both the liver and duodenum. A cholecystectomy typically involves one surgeon extending the cystic duct while another applies clips and cuts it. As the surgeon manipulates the ducts or gall bladder, these ducts stretch and slide according to axial linear forces (figure 10.5). To simulate this, we modelled the system as a series of segments with nodes at the junction points. Once a node is grasped, a virtual spring is put in place between the tool and the node. The spring extension forces are transferred to the node which transfers the forces to adjacent nodes, by extending virtual springs within the segments. In this way, each node has knowledge of all the forces acting on it and can reposition itself depending on those forces. The extension of the grasping virtual spring provides force back to the haptic tool.

Nodes can collide with other organs, and can be simultaneously grasped by the remote user. Since the design allows for any number of forces to be directed to each node, both collision forces and tool forces can be accumulated with the spring forces into a resultant force used to reposition the node. A similar system is described in [249]. This mechanism allows us to incorporate the remote user’s actions into the physics model. To avoid temporal inconsistencies, it is necessary to nominate one of the collaborating machines as a ‘physics server’ which collects all these forces, resolves them and calculates the resultant node positions.

Initial tests showed that latency of the network over global distances introduces instability into the system. We were able to overcome this by ignoring any momentum or acceleration. Objects either move under forces or stop when those forces are in

Fig. 10.4: Deforming the stomach.

Fig. 10.5: Two instruments manipulating the gall bladder
balance. We found that, while this may not be suitable for all collaborative environments, for surgical simulation, where objects need to be moved and stretched, but not sent on trajectories, the behaviour was convincing enough to satisfy the participants of the trial.

We also found that this mechanism allowed us to avoid the need to lock objects for editing [49] [96], such as described in [87]. Since all interaction with objects is based on real world actions (pushing, pulling etc), and the duration of these interactions is relatively long compared to the latency of the network, we can accumulate all the forces into a single resultant, and therefore allow both users to interact simultaneously at any time as they might do in the real world.

**Diathermy of tissue**

The gall bladder is attached to the liver with a webbing of tissue. In the operation, this is cut with a diathermy tool, which uses a cauterizing action to separate it. We simulated the webbing as a polygonal elastic membrane as described above for organs such as the liver. This enables it to be haptically felt and deformed. Points on the edges of the webbing are linked to specific points in the gall bladder and liver, so that deformations of these organs are also transferred to the webbing itself (figure 10.6).

The diathermy effect is created by removing individual polygons from the membrane if they are touched by the diathermy tool for longer than a specified time. The diathermy tool also produces bleeding if it accidentally touches the gall bladder or any of the ducts. A touch-sensitive surface triggers this.
Clipping and cutting the duct

Having separated the gall bladder webbing, the next stage in the procedure is to clip the cystic duct ready for cutting (figure 10.7).

If the duct is cut correctly between two clips, the gall bladder and its attached duct segment is detached from surrounding objects and can be removed. If the cut is made in a place that is not bordered by clips ‘upstream’ and ‘downstream’ on the duct, either bile or blood droplets are released.

Breaking the duct

A common mistake is to apply too much tension to the duct before clipping and cutting. We added the capability for the duct to rupture, emit fluid and eventually break if the extension of any segment is too great.

10.2.4 Other scenes

We also provided a virtual white board, a virtual light box to view medical scans and a virtual video player which can show an actual operation recorded by a laparoscopic camera within the body. The haptic buttons and sliders which control these items can be used by both users. The users are also able to touch and draw on the screens of any of these. The haptic touch triggers the flow of ‘virtual ink’ from the tool, so that it can be used much like a ball-point pen or marker.

10.2.5 Communication

We developed this application using the Reachin Core Technology API [180] (formally Magma [221]) which provides haptic and graphic rendering of the scene. Reading and writing network sockets occur through separate, dedicated threads.

We use replicated databases on each machine with update data being transferred between them when necessary using ‘remote routes’ that exist across a network and transfer data with TCP/IP for less time-critical data and UDP/IP for more time-critical data.

10.2.6 Results

The audience consisted of about 70 delegates attending the Health and Medical Simulation Symposium and associated Simulation Technologies Conference. At the end of the session we received 47 completed questionnaires. Ten of these also participated in the haptic interaction.
The interaction between the surgeons was rated as either very good or excellent by 97% of respondents. Eighty-seven percent rated the remote teaching as either very good or excellent. Eighty-nine percent rated the ability to interact with the master surgeon as she manipulated the virtual structures as very useful or extremely useful, and 89% also rated the ability to be guided by an expert surgeon as either very useful or extremely useful. The data showed that 100% reported a high or very high sense of presence with their teacher and 87% engaged highly or above with the scenario.

10.2.7 Further work

We are planning to extend the collaboration to allow the instructor to connect to one of a number of students, working alone or in pairs on a task. The concept is for the remote instructor to be able to join or leave a session at will. The students need not all be co-located. We are also adapting this technology to other surgical scenarios [103].

10.2.8 Conclusion

The demonstration showed that it is possible to overcome the technical difficulties involved in presenting a haptic teaching environment, linking two institutions across the world. The resulting feedback endorses the ideas behind this trial and provides encouragement for further exploration in these directions. It showed that remote demonstration and discussion can use a rich set of interface components and need not be limited to conventional video conferencing technology.
10.3 Discussion

There have been several haptic cholecystectomy training simulators reported in the literature [229], [13], [244], [14]. The work described here differs from these because it is primarily aimed at investigating the use of the technology to allow an instructor to be immersed in the environment along with a student through the use of a network connection. This work focuses on how this can be implemented, how network latency can be accommodated, and whether such a dual-user, mentoring arrangement is an effective use of simulation equipment for training.

Basdogan et al. list the risks associated with gall bladder surgery as; an improperly inserted gall stone catheter, bile duct injury, misidentification of the various ducts and blood vessels, and clipping or cutting the wrong duct [14]. They developed a simulator that addresses catheter insertion and a gall stone clearing method. We have not addressed the catheter or gall stone investigation stage of the procedure. Our effort has been directed towards the procedural issues of identifying the correct ducts, manipulating those ducts, diathermy, clipping and cutting.

Both Basdogan, ibid., and Webster [244] confirm our belief that haptic interaction with simulations of soft tissue is a challenging problem. Basdogan states that it is “a nontrivial problem that calls for prudence in the application of mechanistic and computer graphics techniques in an endeavour to create a make-believe world that is realistic enough to mimic reality, but efficient enough to be executable in real time” [14] (page 273). They address this with a hybrid of a mesh-based finite element model for organ tissue, and a particle based approach for the flexible catheter. In our work, we needed to approximate the physics of tissue behaviour, to accommodate the ability to have networked simultaneous interaction. For this reason, the model for the duct system uses a particle approach, with duct centre-points attached by spring-damper systems. These centre-points collect all applied forces and redistribute themselves according to the force model described in the paper (section 10.2.3), stretching and moving the duct as they go. The algorithm differs from Basdogan’s catheter model in that the forces are being calculated on only one of the communicating machines, which then redistributes the derived position information to both machines’ rendering routines. As both the surgeons manipulate the gall bladder and ducts, the rest of the cystic duct system slides, stretches and moves as a consequence of the applied forces. The magnitude of the
movement is typically significant compared to the diameter of the ducts themselves, so it is important to ensure that their instantaneous position and arrangement matches, on both the networked machines. The iterative collection of forces occurring at each duct node joint and the position redistribution algorithms described in this paper ensure that this requirement is met. However, for the other nearby body organs, such as the stomach and liver, it is sufficient to utilise a more approximate algorithm, since typical deformations are small compared to the overall dimensions of the organs. Also, these organs are not the main focus of this particular simulator. The forces produced by the interaction of the tools are independently used to calculate the deformation of the surrounding surface and these are combined with the deformation caused by the user at the other end of the network. The pliability of the surfaces associated with these body organs introduces a damping component which fortunately assists in achieving a stable dual interaction capability. This solution would not be suitable for dual interaction involving movable rigid surfaces.

Webster comments that “The fundamental trade-off [in surgical simulation] is accuracy versus interactivity. In haptic surgical simulation this problem is especially acute, as the user must feel contact forces (and see the graphical deformations) that are accurate yet computed in real time.” [244] (page 221). He tackles the issue by pre-computing matrices that are stored in a computer file that can be read in at the start of the application. At run time, the correct stored matrix can be retrieved and used faster than it can be calculated. A problem with this is that it does not allow for topological changes, such as cutting of tissue or ducts, since this would require a re-calculation of the matrices, a process which Webster states can take several minutes. Our system accommodates such changes in the structure at the cost of physical realism. However, our clinical advisors considered that such a compromise was outweighed by the benefit of allowing shared immersion between teacher and student. Since we were concentrating on the cognitive skills associated with the procedure, the ability to share a common viewpoint with an instructor or fellow student, along with the capacity to discuss and assist each other with manipulation was seen as more important than providing an environment to accurately teach the detailed sensorimotor skills associated with the operation. These latter skills would require a more accurate reproduction of the tissue compliance and deformation than that provided by our system. Obviously, a combination of the two is a desirable long term aim. Others [188] [93] have followed
the same philosophy. Hayward states that “The characteristics of the human haptic system allow in some cases the use of simplified physical models to render haptic objects that compete in realism with actual physical objects”, ibid, (page 24). Alternative approach has been used by Boulanger [29] who recorded tool forces during real ophthalmic operations, to be reproduced in a training simulator. Approximate methods may not be satisfactory in that area of surgery because of the weak forces required when working on the eye.

The training system has been extensively demonstrated and trialled at events such as the Australian College of Surgeons Science Conference (2003), Stanford University School of Medicine (2003 & 2004), an Internet2 Consortium’s Member Meeting (2004) in Austin, Texas, and a presentation at the Washington D.C. Internet2 Consortium’s offices to the US Federal Communications Commissioner (2004), as well as in system trials between Australia and Sweden. In each of these demonstrations, one end of the system was located in Canberra, Australia and the other end at the event location. Many of the users were medical practitioners, some of whom were surgeons with experience in cholecystectomies. The hand guiding feature attracted much attention, and it was seen as especially novel to be able to ‘touch’ and ‘feel’ another person’s actions half way around the world.

It was discovered that one of the benefits of hand-guiding was the ability to assist a novice in the correct technique of putting the gall bladder/cystic duct system under longitudinal tension. The gall bladder is connected to the cystic duct, which then bifurcates into two ducts – the common bile duct and hepatic duct. The hepatic duct then splits in two before joining the liver at two places and the common bile ends at the duodenum [81]. During a cholecystectomy, the gall bladder/duct system is freed from the liver and then put under tension to facilitate the application of clips and eventual cutting of the cystic duct. Too much tension can result in rupture of the duct and escape of bile fluid or blood. The organs should be manipulated to allow the surgeon a clear view of the area for diathermy and clipping. The hand-guiding feature was found to provide a very useful way of indicating the amount of extension that should be applied. The instructor could ‘grasp’ the student’s hand haptically while the student had hold of the fundus of the gall bladder. Then the instructor could pull the student’s hand in the right direction with the appropriate amount of force while feeling the resistance of the tension in the duct.
During this manoeuvre, the force and position calculations of the system involve several contributing components. The student’s tool is affected by three forces. Firstly, there is the force applied by the student onto the haptic device that is represented by the grasping tool. Next, the instructor’s ‘guiding hand’ tool, with its position being transmitted across the network, exerts a Hookian force on it through an invisible, simulated spring. Finally, it is connected to the gall bladder with another invisible simulated spring, which is created at the moment of the initial grasp action, and which varies in length (and resulting force) depending on the relative movement of the tool tip and the gall bladder.

The force from the connection to the gall bladder is the result of a sequence of interrelated movements of duct segments, as described in the paper (section 10.2.3). The duct segments, (which are each simulated as a spring-damper between two nodes surrounded by a visible tapered cylinder), are moved according to the total of all forces acting upon their end nodes. These movements are sequentially calculated, segment by segment, along the duct, taking into account any bifurcations, until the point being grasped is calculated. This last calculation also takes into account the invisible

![Figure 10.8: Force model of cystic duct system connected to a user’s tool and ‘hand-guided’ from a remote user’s tool](image-url)
‘grasping’ spring to the student’s tool. The repositioning of each node in this system depends on the forces applied to it from any connecting segment as well as any possible tool grasping forces or collisions with other body organs. During each recalculation cycle, each node collects all forces acting on it, computed from the extension or compression of its connected springs (including invisible springs to any user tools at either end of the network connection), as well as its penetration into colliding objects. The resultant force at each node is then calculated and the node is repositioned marginally in the direction of the force, according to the algorithm explained in Chapter 9, section 9.2.8. Since each node handles its own force resolution and repositioning, the complexity of the interconnected system is reduced to a simple iteration through the interconnected chain of segments. Each node’s position is then calculated from only its neighbours’ positions, and the force onto the users’ haptic tools depends only on the positions of their immediate connections and collisions (see figure 10.8).

Since the user’s tool is connected to the grasp point on the gall bladder or duct by a simulated (invisible) spring, there can be a visual discrepancy when a large amount of tension is applied to the spring – a gap appears between the user’s tool and the tissue. The user still feels the force of the stretched tissue, and the duct system behaves as if it is still grasped and being manipulated, but visually the tool does not appear to be in contact with the tissue because the simulated spring has stretched enough to allow the tool to emerge from the tissue surface. This problem could be solved by modifying the graphic representation of the surface to deform always to the grasping tool tip. The underlying physics model and haptic behaviour do not need to change, as only the graphics rendering needs to be adjusted to provide a more believable representation. This work has not yet been done.

10.3.1 Data Flow

During a typical cystic duct manipulation there are three sets of data being transmitted between collaborating machines: model state data which is input into algorithms occurring within the graphics thread, haptic state data which is used in algorithms occurring during the haptics thread and video/audio data which is processed independently on a separate computer to the simulation.

The model state data is transferred at approximately 30 Hz as part of the graphics refresh cycle. This rate is limited by the power of the processor and complexity of the logic that needs processing, and basically, runs as fast as possible. Typically the
transmission rate for this data is 165,840 bits/sec. Table 10.1 shows the components of this.

<table>
<thead>
<tr>
<th>Table 10.1: Components of model state data flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type</td>
</tr>
<tr>
<td>Position vectors</td>
</tr>
<tr>
<td>Orientation vectors</td>
</tr>
<tr>
<td>Floating point values</td>
</tr>
<tr>
<td>Boolean values</td>
</tr>
<tr>
<td>Integer values</td>
</tr>
<tr>
<td>Total bits/cycle</td>
</tr>
<tr>
<td>Typical cycles/second</td>
</tr>
<tr>
<td>Bits/second</td>
</tr>
</tbody>
</table>

The haptics state data is transmitted at 400 Hz. This figure was determined empirically by winding back the value from a starting point of 1000 Hz to discover the minimum value that could achieve a stable interaction over a network with a latency of 290 ms. The rate is governed explicitly within the code within the 1000 Hz haptic cycle, but is one of the configuration parameters for the program. The transmission rate for this data is 165,840 bits/sec. Table 10.2 shows the components which comprise this.

<table>
<thead>
<tr>
<th>Table 10.2: Components of haptic data flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type</td>
</tr>
<tr>
<td>Position vectors</td>
</tr>
<tr>
<td>Total bits/cycle</td>
</tr>
<tr>
<td>Typical cycles/second</td>
</tr>
<tr>
<td>Bits/second</td>
</tr>
</tbody>
</table>
The video and audio data is transferred in either PAL or NTSC [9] format using the DVTS codec [163], (at 25 Hz frame rate for PAL), consuming a total of 30 Mbits/sec in each direction. The model and haptic data therefore, consumes approximately 0.2 MBits/sec, which is insignificant compared to the 30 MBits/sec video transmission rate. Alternate video compression technologies, such as MPEG2 [252] could be incorporated, bringing the total data flow down to approximately 4.2 MBits/sec. Also, the system can be operated quite satisfactorily with only an audio link. Such a configuration consumes a total data flow of approximately 0.25 MBits/sec bringing the system total to 0.45 MBits/sec. Operation with no video or audio link is possible, and would consume only 0.25 MBits/sec, but without it, users had difficulties conveying their intentions and coordinating their actions.

It can be seen from these figures, that such a system would require significant broadband infrastructure to run in the full, video enhanced mode. However, with only an audio link, much lower bandwidth facilities are needed. Users of the system are likely to be concentrating on the 3D model, not the video screen. They are also likely to be wearing stereo 3D glasses. Because of this, the addition of the video link does not enhance the experience greatly, since eye contact is not easily achieved.

The network connectivity to allow this data flow also requires access through firewalls of any institutions hosting the equipment. The method used for these experiments involved obtaining access from a particular host machine’s IP address over a specific port. It was necessary to put this in place separately for both TCP and UDP [220] communication protocols.

10.3.2 Tissue Cutting

The gall bladder is connected to the liver with a webbing of elastic tissue, as mentioned in the paper (section 10.2.3). This is modelled as a ‘web-strip’, comprising a triangulated membrane surface with specialized behaviour along its long edges which allows it to be attached to both the gall bladder and liver. Such a configuration enables it to stretch and move when those organs are moved, as well as flex and deform when the user’s tool interacts with it. The webbing is designed for use where the length of the membrane is greater by a factor of two or more than the width. It consists of a grid of rectangles, each sub-divided into two triangles across the diagonal. The rows of rectangles form bands, each stretching across the width of the strip. The strip can have a series of attachment points along its long edges, where routes from other vector
positions can be directed. These vector positions typically would be selected vertices of the liver or gall bladder. As the particular vertex moves, the route ensures that the corresponding point on the web strip also moves. These attachment points need not be at the end of every web strip band. Unattached band points interpolate their positions between the attached ones. As an attached point moves, the bands surrounding it reposition themselves and the triangles across the band also interpolate to reposition themselves evenly. Initial trials showed that although this scheme worked satisfactorily, the resulting smoothness of the strip lacked the organic appearance of tissue, even when rendered with an appropriate texture. By adding a randomising factor to the interpolation algorithm, a more irregular surface was achieved, providing a much more believable appearance. The graphical rendering of the strip was also modified to give the tissue some thickness, although it could still be specified as a single surface, but with a thickness parameter. The haptic behaviour did not have to take the thickness into account as the tissue was thin enough for the mismatch with the graphical rendering to be imperceptible [40].

The web strip also has the capability to be cut. In the application this occurs when the diathermy tool comes into contact with it. When the tool touches the surface, it triggers an algorithm which increments a counter stored for that particular triangle of the web strip. For each graphics cycle that the contact occurs, the counter is again incremented. When the counter for a particular triangle reaches a certain number (specified by a ‘toughness’ parameter) the triangle is removed from the web, and the user sees a small hole in its place. When at least one triangle has been removed from a band of the web strip, the band no longer has connectivity from one side of the strip to the other. Therefore, movement of any attachment point at one side will have an effect to the cut point, but no further. The interpolation algorithm takes this into account, only interpolating left hand side movements on the left hand part of any cut band, and right hand movements on the right hand part. When the web strip has been cut completely along its entire length, all bands have been cut and the attached gall bladder can be removed, along with its attached half of the web strip, leaving the other half attached to the liver.
Chapter 11.

Accommodating a Class

11.1 Introduction

This paper, *A Remote Interactive Masterclass in Surgery*, was presented at the OzCHI 2004 conference, in Woollongong, Australia in November 2004. It describes work which builds on the surgical training system covered in the previous chapter, to allow the provision of a training environment to a class of students as opposed to the tutoring of one student. It also describes the results of a user study performed at an international teaching event using the technology.

The OzCHI conference is the main computer human interaction conference in Australia. The delegates ranged from having a small to moderate amount of technical background. They were more interested in human factors issues than the technology of haptics and virtual environments. Therefore, the paper was written to give a more descriptive presentation of the teaching system and the audience’s opinions on it, rather than the algorithms within the software.

Contribution: I co-ordinated the technical components of the event while my co-authors were involved in the audience survey, providing anatomical and surgical advice and arranging for a surgeon to instruct the class. I wrote 60% of the paper, comprising the sections on the technology of the stereo anatomy viewing (although I did not develop the original anatomy viewing software – just adapted it for remote interaction), haptic surgical training, networking and the presentation environment. Co-authors wrote the sections on the audience survey, with some input from me (section 11.2.12). This survey was mostly covering the audience response to the work that I developed.
Publication 8: A Remote Interactive Masterclass in Surgery

Chris Gunn  
Duncan Stevenson  
Alex Krumm-Heller  
CSIRO Information & Communications Technologies Centre

Sakti Srivastava  
Patricia Youngblood  
LeRoy Heinrichs  
Parvati Dev  
Stanford University School of Medicine

Abstract

This paper describes a concept demonstration of a remotely conducted, virtual surgical training master class. A networked immersive virtual environment was used to link instructor and selected students, while the remaining audience of health professionals observed both the contents of the instruction and the instructor-student interactions. The demonstration illustrated the multiple levels of computer-human interaction involved in this large-scale scenario. The focus of the interactions between the computer systems and the participants was at the level of the problem domain, so that the participants, both the specific students involved and the wider auditorium audience, would feel that their experience was in some way akin to the instructor being present in person and conducting demonstration surgery. The actual demonstration was conducted at a simulation technologies conference held in Canberra, Australia. The medical and surgical instructors were at Stanford University in the USA. The responses of the audience and participants were collected in a questionnaire and the results are presented.

Keywords


11.2.1 Introduction

Application motivation

Virtual Reality technology is emerging as an important component of modern surgical training. Cosman et al. [56] describe the limitations of the current apprenticeship model of training, including reduced exposure to an adequate spectrum of operating procedures
and the need for a modern adult-education model of training with “continual high-quality feedback on performance” [56]. They discuss the advantages and disadvantages of current work in this area and conclude that “There is no doubt that simulators will play a role in the training of future generations of surgeons”. Powerful evidence that virtual reality technology does significantly improve the operating performance of resident surgeons was recently demonstrated in a randomised double-blinded study carried out at Yale University [203].

In countries like Australia, where the population is small and unevenly distributed, access to specialist surgical expertise for training can be difficult. Surgical residents commonly travel long distances to attend specialist training courses, at significant expense and disruption to their professional and personal lives. The demonstration described in this paper explores the concept of combining remote interaction with an experienced surgeon instructor and the best attributes of virtual reality technology for this class of training.

11.2.2 Networked virtual environments

Collaborative (networked) virtual environments are expected to have many benefits for training at a distance [211], especially for skills training in the context of surgical procedures and supported by appropriate simulation capability. Others have shown [15], [196] that haptic feedback enhances performance in a collaborative virtual environment, but it is known that haptic feedback makes the issue of network quality, especially latency, a problem of prime importance [145]. The basic architecture for the collaborative virtual environment used in this demonstration was developed within CSIRO over the period 2000-2002, and was presented at the ACM SIGGRAPH conference [84]. It deals specifically with network quality and the issues of latency in the haptic interactions involved, between the two collaborating virtual environments.

This demonstration extends that work significantly in the directions of the quality and usability, and to include wider audience involvement. For this demonstration a temporary broad-band internet connection was established between the two venues, a conference auditorium in Canberra, Australia and the Stanford University School of Medicine in the USA.
11.2.3  The computer-human interfaces

Three pairs of computer-human interfaces were involved in this demonstration. The anatomy instructor presented his lecture to the whole audience who observed live video of the instructor with his real-time, pointer-based commentary on a large-format, stereo display of images of cadaver dissections. The surgical instructor and student worked with an immersive virtual environment system containing a rich set of multi-modal display and interaction components. The body organs were represented by simulations with attributes roughly approximating real tissue, having the ability to be compressed and stretched by the participants’ instruments. These components formed the interface for each of them to their own virtual environment and, through the network, linked them to each other. The audience also viewed the surgical instructor and student at work, live at one end and through streaming video from the other, together with a real-time large-format stereo display of the 3D surgical model and their interactions with it.

The intention for each of these three computer-human interfaces was that the participants would be able to engage directly with the teaching content being presented, that they would also feel a sense of engagement with each other, and a sense of each other’s presence. The success of the demonstration would be measured by the extent to which the participants felt a significant level of engagement and presence, and by the audience’s perception that this goal was achieved.

11.2.4  The Master Class Demonstration

The demonstration used two scenarios. The first was an anatomy instructor describing anatomic structures relevant to the surgical exercise that followed, and the second was a surgical instructor, also located remotely from the venue, teaching a surgical trainee or student about conducting a related surgical procedure. Both the surgeons and the audience observe the anatomy lesson, and the audience watch the interactions between the surgical instructor and each student, and also observes the data, models or objects that were used as part of the teaching, in the style of an audience watching a master-class. The concepts being explored included:

- An anatomy instructor, located remotely, displaying unique stereo images to inform in an engaging manner the surgical instructor, the student, and the audience about the anatomic region of surgical interest;
• The student and surgical instructor engaging in a meaningful dialogue both with words and with actions, mediated by virtual tools, objects and interaction interfaces;

• The audience being able to observe the dialogue and the subjects of the dialogue in a comfortable and natural manner;

• The geographical distance between instructor and student/audience (actually approximately 12,000 km) being no barrier to successful interaction.

The first part of the demonstration involved an anatomist at Stanford (Dr Srivastava) presenting the relevant surgical anatomy using large-format 3D (stereo) images of cadaver dissections. He controlled the display from Stanford and used pointers to features in the images to support his narrative.

The second part involved a surgical scenario — the removal of the gall bladder (cholecystectomy). This scenario had many attributes that were appropriate to this concept. The anatomy of the abdomen is complex but structured (organs are in front, behind, under, connected to other organs) and well suited to the spatial teaching capabilities of the virtual environment being used. The sequence of surgical steps is well-defined (identify, lift, grasp, pull, hold, diathermy, clamp, cut, remove) and can be taught with reference to an anatomical model. There is scope for responses between instructor and student (“find this organ”, “what is that organ”, “what is connected to this organ”). Several of the steps are collaborative, with one person holding and stabilizing while the other cuts or clamps.

11.2.5 Remote anatomy lesson

The juxtaposition of anatomy images and corresponding surgery video clips or simulations has been used very effectively in remote teaching situations. In October 2002 the California Orthopaedic Research Network (CORN) presented an interactive demonstration containing real-time streaming video from the operating theatre, stereo anatomical images of the relevant anatomy, real-time commentary from observing surgeons and real-time questions from students, all located at a number of separate sites across the USA. An audience of approximately 200 attendees at the Internet2 Fall Workshop watched the various data streams under moderation by a panel of clinical and networking specialists [67]. The inclusion of a remote presentation of anatomy in this concept demonstration draws on the experience of this earlier work.
The image set used for the anatomical part of the demonstration was taken from the famous Bassett collection [16]. The Bassett collection is a unique set of more than 1500 stereo images that consists of detailed anatomical dissection of human cadavers in all anatomical regions (from head to toe). David Bassett conducted the painstaking dissections over a period of several years in the 1950’s and 1960’s. William Gruber, the inventor of the 3D View-Master meticulously photographed these dissections (figure 11.1). The high quality of these images was an important feature in selecting them as a part of this multi-component session.

Eight images relating to the particular surgical procedure were chosen and an anatomy lesson plan was prepared to complement the virtual environment component. From the audience’s point of view there was no indication, in the way it was presented, that the anatomy lesson was coming live from the other side of the world. Tight timing of the overall timeslot within the host conference prevented a question-and-answer dialogue between the audience and Dr Srivastava but in other circumstances this would have been the natural way to conclude this component. The anatomy lesson was presented using a networked application called the Remote Stereo Viewer which can act as a client, retrieving the images from a remotely located data base [66]. It was tailored to meet specific security requirements for the temporary broadband network connection between Canberra and Stanford.

The Remote Stereo Viewer was developed by Dr. Steven Senger at the University of Wisconsin, LaCrosse. (http://www.visu.uwlax.edu/NGI/RSV.html) Important collaborative features that require multicast could not be used due to network limitations across the Pacific.

11.2.6 Remote interactive explanation of surgical procedure

The surgical procedure instruction used a collaborative simulated surgical training environment developed at CSIRO and described by [85]. It is based on the CSIRO
Haptic Workbench (figure 11.2), a desktop immersive haptic virtual reality system that is well-suited to simulated tasks that are contained within a person’s arm’s reach [221].

The surgical instructor led students through the simulated procedure, while experiencing a 3D scene comprising liver, stomach, gall bladder, kidneys and other abdominal organs, all of which could be manipulated by the participants. The system continuously transmitted incremental changes in the 3D model (anatomy, instruments, pointers and annotation) between Canberra and Stanford, keeping all movable components, as well as the users’ instruments, synchronized with each other. It allowed both participants to simultaneously draw within the 3D scene while discussing the anatomy, and cooperatively grasp pliable body organs (figure 11.3). They could also cut tissue, clip and cut ducts (figure 11.4) whilst feeling the actions and forces provided by each other across the Pacific Ocean. Using the haptic capability of the system, the instructor could grasp the student’s instrument (figure 11.5a), and guide it to the correct part of the

Figure 11.2: The CSIRO Haptic Workbench

Figure 11.3: a) Manipulating a simulated gall bladder. b) Annotating in 3D.
They were both able to draw diagrams on a virtual white board as well as annotate a virtual medical scan-viewer. This permitted detailed discussion about the techniques required in the gall bladder removal operation. Each was able to point and sketch questions and answers, adding to the flow of information between instructor and surgeon.

A virtual video player (figure 11.5b) embedded in a separate part of the scene, allowed the participants to remain immersed in the virtual environment while they viewed a pre-recorded video of real surgery. Each participant could operate the video and draw on the virtual screen while discussing the operation depicted. The virtual video players at each end of the network connection were synchronized so that each participant saw, and was commenting on, the same video frames. This video feature helped to bridge the gap between the diagrammatic, abstract presentation of the surgical procedure delivered using the virtual anatomical models, and the reality of actual surgery. Its role parallels the blend of streaming surgical video, static anatomical images and off-line discussion that was presented at the CORN demonstration mentioned above [66].

The initial demonstration of the surgical procedure had an Australian surgeon, Dr Patrick Cregan, role-playing a surgical trainee. The instructor, located at Stanford University was Dr Sherry Wren. Following this, several members of the Australian
audience participated in the surgical procedure lesson, one at a time, under remote direction from Dr Wren at Stanford.

11.2.7 Audience involvement

The audience played two roles. At the local level they watched and listened, observing the clinical teaching points being made and observing the way a student might respond to an instructor in such a situation. Their feedback would point the way towards an environment that might maximise the value to be gained from a highly expert but very busy surgical instructor.

Their other role was perhaps more important. Most of them were health professionals or worked in health and medical education. They could, therefore, comment with considerable authority on the value and relevance of the concepts being presented. At the end of the demonstration the audience was handed a questionnaire to gather their impressions and personal responses to what they had experienced.

11.2.8 The presentation environment

In Canberra the conference presentation environment, shown in figure 11.6, used high quality video, echo-free audio, graphical 3D interactive models of human body organs shared over the network, bi-directional haptic interaction and 3D stereo visualization. This involved several discrete sub-systems – each of which was achieved in both Canberra and Stanford by different combinations of hardware and software. The entire system made use of a high bandwidth connection between the two sites, via a link involving three academic and research networks; CeNTIE [48], AARNet [1] and Internet2 [106].
To ensure that all views were continuously visible from the audience and that all conversations by participants could be easily heard we mounted a camera and microphone above two of the display screens. This had the effect of allowing face to face conversations while the audience looked on.

At Stanford a single camera and screen were used to record and display the video. A single microphone was used to record audio and headphones were used to play audio. The use of headphones meant that an echo canceller was not required at the Stanford end to prevent any echo being heard in Canberra.

A single video/audio stream was sent in both directions between Canberra and Stanford. A high bandwidth connection allowed the use of Digital Video (DV) over IP transmission, providing broadcast quality video and audio. There was no jerkiness or flicker in the video and very little latency – allowing a natural flow of discussion between the participants.

In Canberra two cameras were used, one giving a close-up view of the surgeon and the other a wider view of the auditorium. An operator controlled which video streams were sent to the local display screens and to Stanford. In addition to the two large plasma displays for the audience there was a miniature video screen built into the workbench to

Figure 11.6: The auditorium layout.
provide a more intimate face-to-face communication between the instructor and the local surgeon.

Audio capture was achieved by two microphones, one on the CSIRO Haptic Workbench and the other at a podium for the speaker. The local audio was mixed and the resulting output transmitted to Stanford. The local audio mix and the audio stream from Stanford were then broadcast locally on loud speakers for the benefit of the audience. An echo canceller was used in the lecture theatre in Canberra to allow the use of microphones in conjunction with loud speakers without an echo being heard in Stanford.

**11.2.9 Information flow between the “surgical trainee” in Canberra and the surgical instructor in Stanford**

This link between the trainee, or student, in Canberra and the instructor in Stanford demonstrates the purpose and intent behind the architecture of the CSIRO Collaborative Haptic Workbench upon which this system was built. The intention was to provide a variety of ways for the two people to communicate about the data in front of them and about the tasks to be performed, and to provide it in such a way that the actions of each are natural in the context of the scenario at hand. The real-time two-way flow of information (voice, gestures and actions upon objects) is intended to support a natural dialogue that combines references in 3D space and time with actual spatial operations (“here”, “that”, “under there”, “like this”, “let me show you”) where the individuals physically move their hands and the virtual instruments that they are holding as an integral part of the dialogue.

The system designers expected that people coming fresh to this system would find it quick and easy to understand what they needed to do. This expectation was met at this event. The members of the audience who took part in the latter stage of this event were all able to use the components under remote instruction with a minimum of explanation.

**11.2.10 Underlying research agendas**

This event was jointly presented by teams from the CSIRO Information & Communications Technologies Centre in Australia and from the Stanford University Medical Media and Information Technologies project in the USA.

**CSIRO Australia**

The CSIRO team for this event is under the umbrella of the Centre for Networking Technologies for the Information Economy [48]. This Centre has established a
broadband research network within Australia containing both long-distance and metropolitan segments, and is conducting research into novel networking technologies and applications. This includes demonstrating and piloting applications in specific fields including a very broadly defined tele-health field. The research looks at the impact of these novel applications, both in terms of the application domain and of how the human-computer interactions (and human-human interactions mediated by the computer and network) evolve.

**SUMMIT Stanford, USA**

The team at Stanford are also exploring what can be done over the next generation of Internet connectivity in terms of collaborative multi-media support for medical and other clinical forms of education and training. Being based in the Stanford University School of Medicine they have a strong primary focus and expertise in medical and clinical education [225]. Their work is supported by a research grant from the US National Library of Medicine.

### 11.2.11 Implementing the demonstration

This demonstration required access to capabilities that are still very much in the research domain. Creating the required network capacity and reliability half-way around the world involved three network research entities (CeNTIE [48], AARNet [1] and Internet2 [106]) as well as significant “last mile” implementation. The Haptic Workbench is still a research tool which needed to be installed and tailored at both ends. The collaborative haptic software required significant modification and extension to support the surgeons’ requirements for a meaningful demonstration lesson. The ability to clip the cystic duct and the use of diathermy to dissect the membranes was added at the suggestion of the Stanford surgeons. They also suggested that it would greatly benefit the teaching utility of the system if error conditions could be represented. Accordingly, the cystic duct and gall bladder were modified to leak bile fluid if they were mistakenly cauterized. Also the cystic duct would rupture, leak fluid and eventually break if it was put under too much tension. This allowed the students to learn the limits of safe practice through error and repetition. The surgical simulation system ran on a dual processor 2.4 GHz PC running Windows 2000.

The Digital Video (DV) format which was used in this demonstration offers broadcast quality audio and video. While its compression ratio is not as high as other common formats such as MPEG2 it is suitable for real-time compression on a commodity
computer. Software developed within CSIRO, based on work by Ogawa et al. [164] provided both video compression and conversion of the compressed video into network packets for transmission over the broadband connection between Canberra and Stanford. The video/audio system ran on its own PC.

It was felt that audio and video quality of this level was important. The demonstration team felt that artefacts attributable to a low-quality link, such as frame freeze, lack of synchronisation between video and audio and poor quality audio, would seriously interfere with the participants’ experience of the content of the demonstration.

Firewall access also needed to be opened up between the collaborating machines to allow an unhindered flow of data packets between them. This required the cooperation of the various network administrators.

11.2.12 Assessment of the event

General assessment

A major aim of the demonstration at the Health and Medical Simulation Symposium was to provide a knowledgeable audience with direct experience of remotely-conducted surgical training supported by high-quality broadband connectivity, complex multi-modal interaction and large-scale modern stereo display and then to seek their feedback. Members of the audience observed very closely the interactions between the surgical instructor and student in a role-play that contained genuine surgical teaching content and demonstrated in real time the content-rich dialogue between the surgeon and student, knowing that the two were actually physically separated by intercontinental distances. Some members of the audience also took their turn to experience, at first-hand, the remote training session. A one-page questionnaire with structured and free-form responses was used to gather audience feedback. The demonstration team also made informal observations of the audience’s response.

The audience consisted of about 70 delegates to the Health and Medical Simulation Symposium and the associated Simulation Technologies Conference. Most of the audience were clinical practitioners (doctors, medical specialists, nurses and medical educators) who already had extensive experience in using simulation technologies in their professional work. They were, therefore, an ideal group to critique our work.

The conditions of the demonstration were designed to maximise the interaction between the participants and the networked computer systems. There were large, high-quality
visual displays, both of the anatomical data under discussion and of the remote instructor. The audio quality was excellent and the stereo display was achieved through light-weight comfortable glasses and high-quality projector and screen. The contents of both the surgical anatomy lesson and the surgical procedural lesson were relevant and meaningful to the audience.

The first informal observation, made by both our project team and the conference organisers, was that the audience was strongly focussed on the event. Throughout the 20 minutes of the actual teaching demonstrations the audience was still and silent and their body language indicated a strong level of attention. At the end of the session we received 39 completed questionnaires. We also received responses to additional questions from the eight people who volunteered to personally experience the surgical procedural lesson under remote mentoring from the surgeon at Stanford.

**Technical Performance Ratings:**
The first three items of the questionnaire addressed the quality of the technical performance of the systems used in the demonstration. Audience members were asked to rate the following features on a five-point scale (1=Unacceptable, 5=Excellent):

1. Interaction between the surgeon and the student (rapport, dialog, etc.):
2. Display of 3D images of the anatomy during the Anatomy Lesson

Thirty-nine members of the audience completed the one page questionnaire, which yields a response rate of 56%. The data in Table 1 show that 97 percent of those who responded rated the interaction between the surgeons as either very good or excellent. Eighty-five percent rated the visual display of 3D anatomy as very good or excellent, and 87 percent rated the remote teaching as either very good or excellent. Table 11.1 shows the distribution of responses.
Usefulness for Learning a Surgical Procedure:
The next three items on the questionnaire addressed the effectiveness of the unique components of the demonstration in terms of usefulness for learning a surgical technique or procedure. The audience members were asked to rate their perception of the following on a 5-point scale [1=Not at all useful, -5=extremely useful]:-

1. Viewing 3D stereo images of the relevant anatomy.
2. Interacting with a master surgeon as she manipulates the virtual structures and uses the 3D drawing tool to explain concepts
3. Being guided by an expert surgeon as the student manipulates the virtual anatomical structures

The data from this set of questions show that 88 percent of those who responded rated viewing 3D stereo images of the relevant anatomy as very useful or extremely useful. Eighty-nine percent rated the ability to interact with the master surgeon as she manipulated the virtual structures as very useful or extremely useful, and 89 percent also rated the ability to be guided by an expert surgeon as either very useful or extremely useful. Table 11.2 shows the distribution of the responses.

<table>
<thead>
<tr>
<th>Technical Performance</th>
<th>Unacceptable</th>
<th>Poor</th>
<th>Acceptable</th>
<th>Very Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score %</td>
<td>Score %</td>
<td>Score %</td>
<td>Score %</td>
<td>Score %</td>
<td>Score %</td>
</tr>
<tr>
<td>Surgeon-Student Interaction</td>
<td>0 0%</td>
<td>0 0%</td>
<td>1 03%</td>
<td>23 59%</td>
<td>15 38%</td>
</tr>
<tr>
<td>3D Images of Anatomy</td>
<td>0 0%</td>
<td>0 0%</td>
<td>6 15%</td>
<td>16 41%</td>
<td>17 44%</td>
</tr>
<tr>
<td>Remote Teaching</td>
<td>0 0%</td>
<td>1 03%</td>
<td>4 10%</td>
<td>23 59%</td>
<td>11 28%</td>
</tr>
</tbody>
</table>
Table 11.2. Audiences rating of “usefulness for learning”

<table>
<thead>
<tr>
<th>Usefulness for Learning</th>
<th>Not At All Useful</th>
<th>Not Very Useful</th>
<th>Useful</th>
<th>Very Useful</th>
<th>Extremely Useful</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Score</td>
<td>%</td>
<td>Score</td>
<td>%</td>
<td>Score</td>
</tr>
<tr>
<td>Viewing 3D Images</td>
<td>0</td>
<td>0%</td>
<td>0</td>
<td>0%</td>
<td>5</td>
</tr>
<tr>
<td>Interaction with Master</td>
<td>0</td>
<td>0%</td>
<td>0</td>
<td>0%</td>
<td>4</td>
</tr>
<tr>
<td>Guided by Master</td>
<td>0</td>
<td>0%</td>
<td>0</td>
<td>0%</td>
<td>4</td>
</tr>
</tbody>
</table>

Ratings of Presence and Engagement:
Eight members of the audience accepted the invitation to try the immersive surgical training system for themselves. We asked each of them to respond to the following about their experience:

1. The extent to which they felt that the teacher was “present” with them
2. The extent to which they felt engaged with the teaching scenario

The data showed that 100% reported a high or very high sense of presence with their teacher and 87% engaged highly or above with the scenario.

Table 11.3. Audience participants’ rating of their experience of the collaborative CSIRO Haptic Workbench

<table>
<thead>
<tr>
<th>“Presence” of teacher</th>
<th>Very low</th>
<th>Low</th>
<th>Acceptable</th>
<th>High</th>
<th>Very high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>5%</td>
</tr>
<tr>
<td>5</td>
<td>62%</td>
<td></td>
<td></td>
<td></td>
<td>3%</td>
</tr>
<tr>
<td>Engagement with scenario</td>
<td>Very low</td>
<td>Low</td>
<td>Acceptable</td>
<td>High</td>
<td>Very high</td>
</tr>
<tr>
<td>Score</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>1%</td>
<td>12%</td>
<td>3%</td>
</tr>
<tr>
<td>3</td>
<td>38%</td>
<td></td>
<td></td>
<td></td>
<td>4%</td>
</tr>
</tbody>
</table>

11.2.13 Conclusion
The concept demonstration of a remote surgical master class showed that it is possible to overcome the technical difficulties involved in presenting a hands-on teaching environment, in real time, linking two institutions on either side of the world. It also showed the feasibility of providing education through demonstration, to a knowledgeable audience and getting structured feedback from the audience members.
Key features of the master class demonstration included linking instructor, student and audience through a range of interface components. The resulting feedback validates the ideas behind this concept demonstration and provides encouragement for the development teams to further explore these directions. The systems trialled here can be configured in a number of arrangements, from a one-to-one, mentoring environment up to a presentation to an audience in a large auditorium. It showed that remote teaching, demonstration and discussion can be carried out using a rich set of interface components and need not be limited to conventional video conferencing technology.

11.2.14 Acknowledgements

The Stanford team, in the United States, was supported in part by NLM/NIH Contract N01-LM-3-3512 to Parvati Dev, PhD, SUMMIT Laboratory, Stanford University.

The convenors of the Symposium, Dr Patrick Cregan and Dr Brendan Flanagan, encouraged the inclusion of this demonstration event in the Symposium program and Dr Cregan played an active role during the demonstrations.

Broadband connectivity between Stanford and Canberra was achieved with valuable assistance from AARNet [1] and from the GrangeNet [80] project.

JumboVision International Pty Ltd [110] provided the 3D projection system and stereo glasses which enabled the audience to fully experience the demonstrations.

=================================================================================================
11.3 Discussion

The demonstration of this teaching event grew out of a desire to investigate whether the haptic surgical simulation environment, described in earlier chapters, could be expanded to involve a class of students. Since the haptic equipment is expensive, it is likely to only be economically viable if several students can be accommodated in either one session, or at least closely monitored individual sessions. The event was both a trial and a demonstration. It was conducted at a medical conference, and attracted more participants than would normally be expected in a surgical class. However, eight of the 70 or so attendees role-played actual students and took part in the haptic interaction, while the rest of the audience viewed in 3D and followed the participants’ progress.

With the software and infrastructure in place for the event, it was realised that the teaching demonstration would be enhanced by an anatomical lesson on the region of interest before the actual surgical simulation took place.

11.3.1 Remote pointing at 3D anatomical images

Section 11.2.5 of the paper describes the viewing of stereo anatomy ‘still’ images by the class, with accompanying description and image control from an instructor located remotely. The instructor provided an explanation of each image and controlled the sequence of images being displayed. He also used a simple mouse-controlled cursor to point to various structures and features as they became relevant. My contribution to this involved the installation of a 3D projection system suitable for a large audience in a lecture theatre and introducing the capability to control and point at the images from a remote desktop environment, using the VNC utility [185]. The image viewing application was running on a computer located with the class and the instructor used VNC to control and view the output remotely from California. He also had a view, via a video stream, of the actual stereo screen that the class was viewing. This acted as a confirmation of what the class was actually viewing at any given moment, since the VNC system had occasional delays in the transmission of the images, although transmission of the cursor was quite reliable.

As mentioned in the paper (section 11.2.5), the anatomical images were produced in 3D using a passive stereo polarised projection system. This required the audience to wear cheap, lightweight polarising glasses to gain the 3D effect. In the system, the instructor’s pointer is a simple computer cursor, represented in 2D. This then appears to
float on top of all the 3D objects in the scene. Such an arrangement gives the effect that the cursor is hovering on a glass sheet, with the 3D scene behind it, making the act of pointing at objects slightly ambiguous. While a graphic pointer moving in 2D on a 2D image is relatively intuitive, a pointer moving in 2D over a 3D scene, while quite useable, introduces some inconsistency.

It should be possible to extend the system to allow the cursor to appear to move in 3D – i.e. moving in, around and behind objects. One method for doing this would be to create a depth map for each of the images – this would provide a Z-buffer, or ‘depth away from the viewer’, for every pixel in the image. Algorithms exist for creating such a depth map from a stereo image source [23] [138]. These depth maps could be computed in advance and stored along with the images. When in use, the program would compare each pixel of the cursor at its position in 3D (i.e. depth in and out of the scene as well as position across it), with the corresponding image pixel’s depth retrieved from the depth map. If a cursor pixel was closer to the viewer than its corresponding image pixel, the cursor pixel would be rendered in place of the image pixel. If an image pixel was closer, it would be rendered in place of the cursor pixel. To enable this to happen, there would need to be a way of controlling the cursor position in 3D. Any one of the numerous 3D pointing devices available could be used for this.

11.3.2 Haptic participation

During the event, students from the audience were able to participate in the haptic interaction, one-on-one, with the instructor, while the rest of the audience were able only to observe and interact verbally. In the system as described, the large screen passive stereo display shows a duplicate of the CSIRO Haptic Workbench’s active stereo (shutter glasses) display, driven via an active-to-passive converter box. The protocol of the event involved each student coming forward in turn, to the single haptic workbench on the stage to participate in the simulation with the remote instructor. Although most of the audience appeared to be interested in observing each other’s performance during each individual session, the sequential nature of the turn-taking was somewhat slow and unwieldy.

In practice, a surgical training session is likely to involve far fewer students than the 70 or so at the event. With a handful of students, it should be possible to provide a haptic device for each student and incorporate the large-screen passive stereo display into the haptic interaction instead of the workbench. In this mode, students could quickly pass
control of the virtual instruments to each other, resulting in a more rapid turnaround of practice and instruction. As a step towards this goal I adapted the surgical simulation system to display its 3D image solely via a passive stereo projection system, deriving its signal directly off a dual-headed computer graphics card. This involves reconfiguring the software to direct its left eye view to one of the graphic card’s outputs and the right eye view to the other. Each output is then fed to a separate data projector, mounted one on top of the other and carefully calibrated to have coinciding pixels on a mono display. The data projectors then shine through orthogonally polarised glass filters onto a polarisation-conserving reflective screen. With this modification in place, the user can elect to either view the 3D image close up, in active stereo on the haptic workbench, or alternatively interact haptically with the objects displayed in the distance on a large projection screen. Obviously, in the latter case there is no co-location of the haptic touch point with the visual representation. Although this is less than ideal [221], it does provide a convenient way of letting a class of people swap control rapidly, and the human brain seems to adapt fairly readily to the mismatch. Such a trade-off may be worth-while in a teaching scenario.

I have subsequently implemented an intuitive way of allowing two haptic devices to ‘swap’ the virtual instruments that they control, and this is described in Chapter 12. Such a swapping mechanism could be incorporated as an easy way to pass virtual instruments on from one student to the next.

### 11.3.3 Video Synchronisation

Section 11.2.6 describes a collaborative virtual video player that was provided as part of the 3D surgical simulation scene. Within the virtual environment, users can swing their view around, or alternatively jump directly to a part of the 3D space that contains a model of a virtual video player, incorporating buttons for play, stop and pause. The player can show a video file stored independently on each of the two participating machines\(^\text{13}\). Typically those machines will run at differing processor and disk access speeds, perhaps resulting in the video on one machine becoming out of synch with that on the other machine. The consequence of this is that the users may make verbal comments to each other on aspects of the video which are not matched to the video

\(^{13}\) Chapter 12 describes an extension to this system which allows the user to select one of a number of video files.
frames that their partner is actually viewing at the time, perhaps leading to confusion or misleading information being conveyed. More significantly, if the video is paused at some point for discussion, or to interactively annotate the screen, it may well be halted at different frames on the two machines, with the users being unaware of the difference.

To guard against this problem, I developed a synchronisation system to ensure that there was always less than one frame’s time difference between the two machines displays. The algorithm for this involved providing each video frame with a sequence number. Each machine sends its currently displayed frame sequence number to the other machine. On receiving a remote sequence number, a machine compares it to its own, and if the received one is less than its own, it reduces its own frame rate by an increment. If the remote sequence number is greater than its own, it does nothing. In this way, the faster machine paces itself to the slower one. The algorithm accommodates any machine that cannot keep pace with a fixed frame rate, and also ensures that each frame is displayed and none are skipped. The latter point is important because in a medical sequence it may be important to observe important transitory events that may appear in only one frame. It is for this reason that a slower machine cannot simply jump forward to synchronise with a faster one.

11.3.4 Enabling Natural Conversation

Section 11.2.8 mentions that a microphone and speaker were mounted in close proximity to the plasma video displays that were used for face-to-face verbal interaction. It is a common mistake, in video conferencing environments, to place microphones on a conferencing table, but to have a video conference screen at one end of the table, or worse still, against a wall away from the table. During the trials of this work that involved distance interaction with video systems, I have observed that human nature causes people to turn and face the person (or image of the person) to whom they are talking. It is such a natural thing to do that, even if a person is aware that the microphone may be in a different direction from the image of the person, they find it very hard to break the habit. A user at the end of a conferencing table is likely, therefore, to turn away from the table and talk to a screen mounted on a wall, even if the microphone servicing that person is on the table in front of them. This can result in a diminished, or perhaps inaudible, representation of the conversation to the remote participant. The placing of the microphones and speakers in close proximity to the screens in the described system, therefore, was important in achieving a natural and
intuitive verbal interaction environment, providing the capability for a conference
delegate to walk up to the plasma screen showing the remote instructor and have a face
to face discussion. The CSIRO Haptic Workbench had a miniature video screen and
microphone attached to it in direct view of the user, for the same reason.

11.3.5 Private Communication Channel

My experience with this event indicated that, when coordinating a public event
involving remote participants, it is vitally important to provide a private communication
channel between the two ends. This is needed to send last minute instructions, such as
requests to mute or un-mute microphones or switch on cameras. Also, when the event is
underway, it is helpful to have an audiovisual controller in each location, monitoring
microphone volumes and perhaps switching video streams as necessary. A private
communication channel between the two controllers greatly assists in producing a
smooth presentation. Such a channel could be incorporated into the overall video and
audio transmission system in use, or could be as simple as a sustained phone call.

This chapter described a surgical training system and a public demonstration of its use.
The next chapter provides evidence for the benefits of such a system through its report
on a clinical study based around a related application, built using that same framework.
Chapter 12.

A Clinical Study

12.1 Introduction

The previous two chapters describe a prototype application providing a dual, immersive, haptic simulation environment for surgical training, and the demonstration of its utility in both a one-on-one training environment and a class scenario. My colleagues and I then decided that it was necessary to use the knowledge we had gained from this research to build a training simulator designed to be commercialised and used in an ongoing basis in a clinical environment. With the assistance of a prominent otologist from the University of Melbourne we were able to develop a surgical simulator directed at procedures on the temporal bone, which encloses the inner and middle ear. This area of surgery was seen as suitable for simulator-based teaching because the specimens that the current, cadaver-based training methods required were difficult to obtain in suitable numbers. Also, the dimensions of the surgical region were appropriate for the haptic equipment and the surgeons typically used a combination of haptic, visual and audio cues to guide them in the task.

The development of this application was a team effort, with my contribution being:

a) the introduction of two-handed haptic interaction and networking this across two machines, thus permitting up to four haptic tools within the scene,

b) the ability for users to swap control of virtual instruments between their left and right hands or between each other through a touch sensitive interface and

c) providing a microscope view of the workspace while maintaining the real-world dimensions of the haptic interaction.
d) bone dust simulation and suction

e) a haptic 3D tool and scene selection user interface

I played no part in the algorithms and coding required for the voxelated model of the temporal bone itself, nor the erosion of this bone with the virtual drill pieces. My contribution to the research project was approximately 40% of the total, but my contribution to this particular paper was approximately 25%. My components of the application that are not mentioned in the paper are covered in the discussion section of this chapter.


The visual appearance of the scene within this simulator is markedly different to that of the gall bladder simulator described in earlier chapters. However, much of the underlying algorithms and code are in fact common between the two programs. One of the design goals of our team has been to develop software which is general enough to be applicable to a variety of surgical scenarios, and in fact also applicable to domains outside of medicine. This application demonstrates the outcome of this goal.
12.2 Publication 9: A Networked Haptic Virtual Environment for Teaching Temporal Bone Surgery

Matthew HUTCHINS, Duncan STEVENSON, Chris GUNN and Alexander KRUMPHOLZ
CSIRO ICT Centre, Australia

Stephen O’LEARY,
University of Melbourne Department of Otolaryngology, Australia

Abstract.

This paper describes a computer system for teaching temporal bone surgery using networked CSIRO Haptic Workbenches. The system enables an instructor and student to collaboratively explore and drill a volumetric bone model including significant anatomical features. Subjective evaluations by otologists have been favourable, and experimental trials are planned.

12.2.1 Introduction

This paper describes a computer system for teaching temporal bone surgery using virtual reality technology. The temporal bone, located in the base of the skull, houses the delicate organs of the middle and inner ear that contribute to hearing and balance. Safe surgical drilling in the region requires excellent knowledge of the complex 3D anatomy, confidence in the correct technique and approach to the procedure, and physical dexterity and tool handling skills. Training in temporal bone drilling is challenging; in particular, it is becoming more difficult to obtain access to the large number of human bone samples required to achieve competence.

From a technologist’s point of view, there is a satisfying match between the requirements of temporal bone surgery training and the capabilities of certain types of virtual reality environments. For instance: stereo microscopy that is similar to current methods of presentation of interactive 3D graphics; tool use that is a good match in workspace and force to currently available haptic devices; the use of rigid bony models that are easier to segment from CT and simulate dynamically than many soft tissues. Several groups have reported on previous and ongoing development of temporal bone trainers [179], [245], [108], [6], [171], [153]. Our system adds to the growing body of research in this area, while at the same time taking a different approach on some of the key problems. Technically, we have chosen a hybrid volume/surface approach to
graphical and haptic rendering of an erodable bone model. In terms of training, we have chosen to implement a networked system that allows a mentor to interactively observe and guide a trainee through the steps of a procedure. Our system is superficially similar to that recently reported by Morris et al. [153], but is unrelated, and differs in the implementation details. Our aim is to demonstrate the effectiveness of this sort of training through experimental trials.

12.2.2 System Overview

Our system uses the CSIRO Haptic Workbench environment [221], with two PHANToM haptic devices from SensAble Technologies [201], and a foot pedal to control drill speed, as shown in figure 12.1. The software was developed using the Reachin API [180], and makes use of our previously reported software tools and approach to developing collaborative haptic surgical training systems [84], [85], [87]. Although the system can be used by a trainee alone, one of our main aims is to explore how networked virtual reality systems can be used for surgical mentoring, both within the same room, and tele-collaboratively over long distances. Thus, the system allows two users at separate haptic workbenches, typically an instructor and student, to share a common view of the procedure, and interact equally.

![Figure 12.1: The Haptic Workbench configured for temporal bone surgery training. (a) Instructor and student seated at networked haptic workbenches. One hand uses a haptic device for drilling. (c) The other hand uses a haptic device for suction, or 6DOF input device to orient the bone. (d) Active stereo shutter glasses generate 3D views. (e) A foot pedal controls the drill speed.](image-url)
Depending on the hardware configuration, there may be as many as four tools active in the scene simultaneously (two hands for each user). As an example of use, the instructor might demonstrate the drilling, then reset the model and observe the student perform the same task. The primary view provided by the system is of a simulated stereo microscope, with a temporal bone model placed centrally (figure 12.2). Several tools are available for selection in either hand, including a drill handle, sucker, facial nerve stimulating probe, marker, eraser, and networked haptic guiding hand. The drill handle can be loaded with cutting or polishing burrs in a range of sizes. The bone models for the system are volumetric, derived from CT scans. We also model, as polygonal surfaces, some of the major anatomical landmarks in the bone, including the dura, the sigmoid sinus, the carotid artery, the facial nerve, the labyrinth, the eardrum and lining of the external ear canal, and the ossicles.

12.2.3 Preliminary Evaluation

Our first significant experimental trials, designed to measure training transfer, will be conducted in November 2004. In March 2004 the system was demonstrated at the annual scientific conference of the surgical specialty in Australia, and at an associated hospital-based temporal bone drilling course, in order to gather qualitative feedback about the acceptability of the system to educators and trainees, and to record suggestions for future improvements. We asked attendees to experience the demonstration and fill out a short questionnaire, ranking the acceptability of various aspects of the system on a five point scale and providing written comments on other areas.

Feedback from the questionnaire was strongly in favour of the concept we presented, and also helped us to identify key ways we can improve the system for future trials. For
example, 54 out of 55 respondents ranked the concept as “acceptable” (3) or better for teaching surgical anatomy, and 51 respondents ranked the concept as “acceptable” or better for teaching surgical planning and approach. 16 out of 16 respondents who experienced remote mentoring across a network, ranked the mentoring concept as 4 or 5 out of 5.

12.2.4 Future Work and Conclusion

Our system has been designed with a strong emphasis on the requirements of teaching safe temporal bone surgery, with less weight given to physical simulation realism. Thus, the initial use will be in teaching surgical anatomy and surgical planning and approach. We hope to soon be able to demonstrate a measurable improvement in surgical technique, at least in the bone drilling lab, after students use our system. This will add value to the existing training, and may enable a reduction in the number of human bone samples that are required to achieve safe competence.
12.3 Discussion

The trial indicated that the virtual reality simulator was viewed very favourably by both students and instructors. According to Fitts and Posner’s three-level motor skill acquisition theory [71], learners initially experience a cognitive stage, where they are intellectualizing the task. This is followed by integrative stage, where the components of the task are co-ordinated and finally an autonomous stage where they are free to concentrate on higher level aspects of the procedure. Reznick and Macrae [183] describe simulator training as being most appropriate for learners in the early, cognitive stages of a training program, where the predominant teaching mode is one of exploration and demonstration. Strom et al. agree [223], finding that in surgical training the learning curve was boosted by early haptic simulation exposure. This training system is particularly focussed on this stage, having the ability to transparently see through the bone to understand the anatomy, practice the stages of the procedure as well as discuss and explore, assisted by demonstrations and guidance from an instructor.

12.3.1 Two handed haptics

Temporal bone drilling typically requires the surgeon to hold the drilling device in one hand and a sucker/irrigator in the other. (The purpose of the sucker/irrigator is to apply water and suck it back in order to remove bone dust and heat from the scene of the drilling.) The need for the sucker/irrigator in the simulator introduced the requirement for two-handed haptic interaction, since the sucker may well make contact with the bone as does the drill, and a haptic response would be expected in both cases. This is complicated by the fact that the reaction of the bone to a touch event by the drill should be different to that when touched by the sucker — when touched by a spinning drill, voxels are eroded. Since this behaviour is encapsulated in the coding of the bone object and not the drill, it is necessary for the bone to ‘be aware’ of which tool is touching it at any time. A two-pass traversal of the scene-graph is used to implement this; one pass for each tool, so that different methods in the code can be activated depending on which pass is current. Each of the two haptic devices can be assigned to one of six possible virtual tools; drill, sucker, facial nerve probe, guiding hand, marker and eraser. When a virtual tool is chosen by the user, a mapping between the haptic device and the tool object is set up. At the start of each scene graph traversal, a flag is stored nominating which (left or right hand) haptic device is being tracked for that pass. Then when a surface contact is detected, the relevant virtual tool is retrieved from the haptic
device/tool mapping. Methods within the coding of the virtual tool use its own tool parameters (such as sharpness, radius etc) to call lower level methods in the contacted object and therefore perform the necessary actions on that surface.

### 12.3.2 Drilling burrs

The drilling instrument can be loaded with a number of different burrs (i.e. end pieces), each having different erosion capabilities when applied to the bone. The burr erosion parameters are passed to the bone along with the drill speed, burr radius and touch point to be used in the calculation of erosion rate and location. There is potential to also track a simulated bluntness parameter on the burr and vary this over time as it is used, although this is not yet implemented. The burrs are selected from a virtual tray by touching the drill handle to the desired burr itself. The burr then animates from the tray into the end of the drill handle with an accompanying click. This departure from reality represents the real-world action of picking up the burr with the other hand and inserting it – done by either the surgeon or an assistant. The animation is a compromise between the real action, and a simple instantaneous swapping of the burr. As a time-saving measure, it proved advantageous to allow the user to touch a new burr with an already loaded drill burr. In that case the current burr animates back to the drill rack, then the new burr animates into the drill handle.

One criticism of this representation of burr loading is that the user touches a burr tray with the currently loaded drill burr. In real surgery it is very bad practice to touch anything apart from a patient with the drilling burr. Such a practice in a simulated environment could be a source of negative training. This failing is not limited to the changing of drill burrs. The user interface for this simulator is contained, along with the virtual temporal bone model, within the 3D environment. When the user needs to change virtual tools they tap a corresponding virtual button with the current tool. Replacements for this interface would need to allow user interface actions that do not involve the touching of virtual tools (especially the drill) onto UI objects. Possibilities include the use of speech recognition (which might simulate asking an assistant to change a tool), an external touch screen UI display (although this may be awkward to view while wearing 3D shutter glasses), or gesture recognition using gestures performed with the current virtual tool (although this may have no correspondence with anything in real surgical practice).
12.3.3 Sharing instruments

During a teaching session an instructor is likely to need to demonstrate various techniques to a student. To allow this, it is necessary to provide an intuitive way for users at each end of the network to share the various instruments (marker pen, eraser, facial nerve probe, guiding hand, drill handle, sucker). Of these, the marker pen and eraser, whilst having simple surface contact behaviour, have no behavioural interaction with the surface nor do they interact with any action coming from the other user. For this reason, it is feasible to allow both users to have these instruments selected simultaneously. However, the other instruments (drill, sucker, probe and guiding hand) can affect actions coming from the collaborating user. For this reason, allowing both users to hold copies of the same instrument is likely to introduce conflicts when they are interacting with the same part of the model. It is exactly this problem that is addressed in the gall bladder simulator and solved using ‘pseudo-physics’ as described in Chapter 9. A similar technique used for this simulator would be more problematical, since the bone surfaces are rigid and therefore would not have the inherent damping provided by the soft tissue surfaces to aid in sustaining stability under dual manipulation. Fortunately, for the temporal bone simulator it was possible to take advantage of the fact that the procedure requires only one instrument of any particular type to ever be used concurrently and the bone itself can be assumed to not flex or move under contact from any of the tools. By also applying these restrictions within the simulator, it is possible to avoid significant complexity.

The method of passing tools between users involves one user placing a tool down and the other picking it up, much as would happen in the real world. The ‘putting down’ and ‘picking up’ action is simulated as a touch sensitive tool tray within the scene. Touching a tool label with a tool will disengage and put down the current tool and pick up the touched one. If a tool is in use, its label will be greyed out and not selectable. Two exceptions to this are the marker and eraser, which are never shown as unselectable, since there can be no conflicts between any simultaneous actions taken with them.

12.3.4 Microscope view

The normal mode of performing this class of surgery is by viewing the region through a stereo microscope. Simulating this is not as straight-forward as scaling the model up or down. This is because a microscope scales the visual representation of the object but does not change the touchable size and shape. The surgeon’s hands and the surgical
instrument still need to move at the same precise scale, regardless of the zoom of the visual representation. To implement this, it was necessary to represent the objects within the microscope twice; firstly for the graphics rendering (with the currently selected scaling to match the zoom of the microscope), and secondly for the haptics rendering, with no scaling applied. Duplication was avoided by including the same geometric model within two branches of the scene-graph; one containing a scaling node in its hierarchy while the other one didn’t have one. During the graphics rendering pass, the thread is prevented from rendering the non-scaled branch and during the haptics rendering pass the thread is prevented from rendering the scaled branch. The bone model is attached below each branch.

This mechanism, however, does not accommodate the various instruments that can appear both outside the microscope as well as within the microscope’s field of view, depending on their movement. Additionally, it is common for an instrument to extend from the view outside to inside the microscope, so, for example, a drill handle may need to be at normal zoom at one end, and zoomed several times at the other. Graphical clipping planes are used to achieve this effect. A graphical clipping plane can apply to a scene graph group node and allows the graphical rendering of its enclosed objects only if they are to one side or the other of a 2D plane in 3D space. Placing these planes appropriately around the microscope prevents the non-zoomed versions of the instruments being seen within. A second branch of the scene-graph contains corresponding clipping planes which face in the opposite directions. This branch contains the zoomable, ‘within microscope’ versions of the same instruments. The effect is that an instrument moving towards the microscope will have any part of itself appear correctly zoomed within the microscope’s field of view. As well as this, any movements within the microscope are also appropriately zoomed. The method causes the ‘disconnect’ between the outside shaft of an instrument and the inside view of it – as occurs within a real-world microscope.

**12.3.5 Bone Dust simulation**

The action of the drilling burr on the surface of the bone results in a build-up of bone dust and debris in the vicinity of the drill. This is simulated through a particle system with the capability to emit particles according to various configurable parameters. These include particle size, creation location, creation rate, velocity and ‘spray’. The spray parameter controls the variability in the speed and direction of the emitted particles.
Without it, they are emitted in a single linear stream. The particles also have the ability to be drawn towards a location in space and to disappear when arriving there. This feature is used in the haptic sucking tool, which draws the particles into its mouth. Because the system is designed for collaboration, it is necessary to replicate any particle generation and sucking on both collaborating machines. However, the accurate replication of each individual particle across the network would constitute an unnecessary use of bandwidth. Since the users would not interact individually with each particle, it proved sufficient to only reproduce the particle emission location, velocity, rate and spray parameters listed above. Such an approximation resulted in the ability to be drilling and creating dust at one end of the network, and sucking the dust from the computer at the other end, with no discernable inconsistencies.

12.3.6 Haptic Properties of the Sucker

The implemented system allows the sucking tool to interact haptically with surfaces in the bone model. Haptic effects from the suction are not currently implemented, but such effects could involve a suction of the tool towards any surface it touches. Since the surface being touched is aware of the touch event, and location, and the tool’s orientation is known, it is feasible to determine the direction of any suction force and vary that with the distance of the tool tip to the nearest surface vertex. Haptic effects from particles entering the tool tip would also be possible but would, in reality, be so slight as to be insignificant.

12.3.7 Software Structure

The software structure used in the temporal bone simulator is the same as that used in the gall bladder simulator described in earlier chapters. The use of the same framework for two apparently diverse surgical scenarios demonstrates its utility and generality. The overall logic and flow of the program is written in Python [177]. The Python code uses the VRML-like Reachin API to assemble nodes into a scene-graph. These nodes are either those provided by the Reachin API itself, or new nodes written in C++. These new nodes are accumulated into appropriate libraries which are then imported as .dll’s into the VRML or Python scripts. For example, the rendering and behaviour for the bone dust, described in section 12.3.5, is encapsulated in the Dust C++ class. This is compiled and included as a member of the Medical.dll library. This library is loaded from within the BoneDrillingApplication.py Python file which is loaded from the Main.py Python file that runs at start-up of the simulator. New simulators can be created
with the same framework by substituting the various components by maintaining the same overall structure.

The next chapter demonstrates that a totally different simulator, one not associated with surgical training, can be created using the same framework.
Chapter 13.

Sculpting in Pairs

13.1 Introduction

The previous chapters describe the development of collaborative haptics for use in surgical training simulators. The benefit that the collaboration technology provides is the ability for an instructor to guide and assist a student in a simulated surgical task, whilst being immersed together in the same virtual environment. In this case the instructor is passing knowledge to the student through gesture, words and demonstration.

It is possible for a similar collaborative virtual environment to be used in other scenarios and in other modes. The next paper, Collaborative Virtual Sculpting with Haptic Feedback, investigates whether the collaborative haptic environment can be used by two colleagues, perhaps equally skilled, to perform an artistic task together. This scenario differs from that of teaching because the two users can choose to work independently or together at different times, rather than one teaching the other. It particularly addresses the question of whether such a dual approach to an artistic endeavour can elicit creativity that would not have been possible otherwise.

The paper appeared in the Virtual Reality Journal, Special Edition on Collaborative Virtual Environments for Creative People in 2006. The journal targets readers who are literate in virtual reality techniques, software and hardware. My contribution to the paper was 100% - I was the sole author and developer of the software. I was solely responsible for the user study and the analysis of its results.
13.2 Publication 10: Collaborative Virtual Sculpting with Haptic Feedback

Chris Gunn  
Commonwealth Scientific and Industrial Research Organisation (CSIRO)  
GPO Box 664, Canberra, Australia  
Email: Chris.Gunn@csiro.au  
Ph: +61 2 62167081  
Fax: +61 2 62167111  
Web: www.ict.csiro.au/staff/Chris.Gunn

Keywords: Haptics, collaboration, sculpting, virtual environments

Abstract

This paper introduces a haptic virtual environment in which two users can collaboratively sculpt a virtual clay model, working from different physical locations connected by the internet. They view their virtual sculpting tools and the clay model in three dimensions; feel the tool’s pressure on the clay as they work, and have their hands co-located with the view of the tool and model. Since the sculptors have independent views of the same logical environment, they can work at different zoom levels, and be in different coordinate systems, even spinning ones, at the same time. This provides them with the capability to explore new styles of collaborative creativity, working off each other’s initiative where appropriate. The system was designed to allow unrestrained, asynchronous behaviour by the collaborating sculptors.

The paper describes the hardware as well as the algorithms behind the deformability of the clay surface and the communications model enabling the distance collaboration. It gives an explanation of a simple conflict resolution mechanism that haptic feedback facilitates and also reports on the results of a qualitative study into the creativity benefits of such a collaborative system.

13.2.1 Introduction

Art can be thought of as a record of human existence. Sculpture is one of the earliest forms of art. From ancient times to our own day, sculpture has represented the thoughts, ideas, emotions and experiences of humanity seen through the eyes of the artist. Sculptors have used their creativity to provide an insight into their interpretation of the characters and events of their times. Historically, sculpture has focused on the depiction of deities, such as the marble gods of Phidas, historical figures, such as the bust of
Caesar Augustus, or the human form, such as Discobolus (the Discus Thrower). Larger works have depicted historical events. These have often been in the form of a relief due to the size of the works and the structural limitations of the materials being used. The earliest historical relief that survives is the one dedicated to Aemilius Paullus in 168 B.C. It depicts the historical battle between the Romans and the Macedonians at Pydna [246].

Societies have long used sculpture as a great leveller, since it does not matter if a person cannot read or write (or even hear or see); they can enjoy it and can try to understand the messages behind the form. In more recent times, sculpture has been used to denote more abstract geometric concepts such as flow, symmetry and juxtaposition. The theory behind sculpture has even been applied to landscaping, architecture and product design.

From ancient times, sculptures have been primarily the work and inspiration of an individual artist. However, many of the larger Greek and Roman works, as well as those of the renaissance, have also included input from one or more apprentices working in a team. Michelangelo spent his early years as an apprentice to the artist Ghirlandio. (He later tried to suppress this fact, implying that he was largely self-taught, perhaps because he did not want to present himself as a product of the workshop system which carried with it the stigma of teaching crafts rather than art [178]).

Right up to the present day, sculpture is perceived primarily as the product of the inspiration and creativity of an individual artist, with the co-operation of any assisting artists being relegated to supportive roles. Traditional (physical as opposed to digital) sculpture has rarely ventured into the realm of allowing two or more sculptors to cooperatively create a work of art and thereby contribute equally to the originality of the activity. If there has been any co-operation between artists in the creative process, it has been a sequential process, with one artist passing on a part-finished work to the other, or two or more artists working on separate components of the whole work [182] [237].

In this paper we describe a system that allows two artists to simultaneously work cooperatively on a common sculpture in a virtual environment. The over-riding goal of the project was to allow unrestrained, individual interaction with the model by two sculptors, without being encumbered by the constraints of their own physical bodies. They would be able to move their hands freely around the model, and interact with it in a natural way, but also have the freedom to ‘pass through each other’s bodies’ in a super-natural way. Artificial constraints imposed by the digital technology were to be
minimized, allowing each sculptor to poke, mould, carve and paint any part of the model at any time. The technology should not get in the way of the creativity, nor stifle the imagination. The aim was to create a co-operative environment where the sculptors could see, feel and interact with each other’s work as it was being created, without being constrained by turn-taking or sharing protocols.

In real-world pottery and sculpture, two artists working on a common work obviously have physical bodies that will limit their actions and behaviour. They will typically collide hands, arms and bodies if they should try to work on a similar area of a piece at the same time. Using a digital, computer-generated, environment we are able to circumvent some of the aspects of the real world that are not to our benefit, and take advantage of those that are. With the use of haptics we can reproduce physical elements from reality when it suits our purpose, and also make use of the non-isomorphic, or ‘magical’ (i.e. non-physically based) capabilities of a virtual environment where it is advantageous to do so, taking advantage of both the strengths and limitations of human perceptions. In [212] Smith describes the trade-off between the ease of learning within a familiar realistic environment against the benefit of having magical abilities to perform tasks with greater ease.

Digital 3D modelling involves using a computer to create a conceptual 3D model which is typically represented as a 2D projection on a computer screen (e.g. Maya, 3DSMax, ZBrush, Renderman and Rhino). Using 3D stereo viewing technology, two slightly differing 2D projections of this model can be delivered to the viewer’s left and right eyes respectively, to create the illusion that they are viewing a 3D shape.

Digital sculpting is a term used to describe a free-form style of digital modelling, where the user directly interacts with the virtual modelling material, rather than with on-screen widgets that may in turn modify the model. Since the user is interacting with the 3D shape itself, hand motions need to also be three dimensional, using some kind of 3D interaction tool [73] [199]. Haptic systems allow users to feel the objects they are working on through force feedback. Haptically enabled digital sculpting applications enable users to feel as well as see the modelling medium as they work. Shaw and Green [206] describe a digital sculpting system which has visual feedback and 3D interaction but does not provide any haptic interaction. Li et al. [131] developed a system which allows the user to sculpt control points which adjust the surface model indirectly. SensAble Technologies [201] market a product called FreeForm® [202] which allows a
user to build up and sculpt a voxel model of a sculpture, assisted by haptic feedback. Nishino et al. [157] describe a collaborative sculpting system that uses implicit surfaces, but has no haptic feedback. None of these systems allow both collaborative sculpting and haptics. The system we developed is a surface-based sculpting system, which has haptic feedback as well as 3D stereo graphics, and also allows simultaneous collaborative sculpting by two sculptors. It was trialled along with an integrated video conferencing facility to allow the collaborating sculptors to communicate face to face.

After a digital model has been created in a virtual environment, the problem remains that it is potentially ‘trapped’ inside the bits and bytes of the electronic system. Although we did not address this problem directly, there exist technologies for ‘freeing’ the model into the real, physical world. 3D printing equipment is now commercially available and can potentially be used to ‘solidify’ the digital models [125]. The system described here allows export to VRML (Virtual Reality Modelling Language) [92], which then allows import into these 3D printing technologies as well as other applications, including web browsers.

13.2.2 Hardware

To reproduce a realistic clay-sculpting experience in a virtual environment we considered it necessary to provide haptic feedback approximating that which a sculptor would expect when working with real material. The interaction point should be co-located with the visual representation of the model. An effective solution to this is the CSIRO Haptic Workbench [221], a hardware configuration which has been previously described for use in mining applications [86] [82] and surgical simulation training systems [84] [85] [103]. It is a frame which places a computer monitor above an angled mirror and allows the haptic interaction to take place behind the mirror where the 3D model is represented (Fig 1). The haptic interaction is calibrated and scaled to match the visual dimensions that are viewed in the mirror. As with surgical simulation, this arrangement reinforces the believability of the virtual system, through the use of
proprioception. The user cannot see their real hands but the virtual representation of the tool they are holding is located in the same place as they would see their hands if the mirror were not there [25].

Force feedback was provided through the use of two SensAble PHANToM 1.5 haptic devices [201], one for each sculptor. These have a resolution of 0.03mm and a maximum force of 8.5 N – a sufficient amount to simulate the pressure required to sculpt clay. Two such systems were connected together through either a local area network or through the Internet.

The virtual model can be oriented in space using a Logitech Magellan space mouse, which is typically used in the non-dominant hand while the user sculpts with the dominant hand. (There is also an on-screen Compass widget that can orient the model in the absence of the space mouse.) Active stereo shutter glasses are used to provide stereo 3D vision.

A miniature screen and spy-cam are attached to the workbench to allow video/audio communication between the two sculptors. This is important to provide more awareness of each other’s presence – users can see and talk to each other as well as see their collaborator’s avatar (in the form of a sculpting tool) in the virtual scene. Such interaction assists in the synchronisation of the users’ actions and decisions on where and what to work on at any given time, and what their intentions are. DVTS software [164] [163] handles the video transmission on Linux machines, providing a very good quality video display, but at the penalty of high bandwidth requirements (~30 Mb/sec). A lower quality video link would likely be sufficient for this application as the concentration of the participants is likely to be more focussed on the sculpted model rather than the face-to-face communications.

A dual processor 3.2 GHz computer with an NVIDIA Quadro4 900 XGL graphics card, running Windows XP provided the haptics and graphics processing at each end. A single processor 1.5 GHz computer running Linux was added, to provide the optional video conferencing capability.

13.2.3 Collaborative Virtual Environments

The term ‘collaborative virtual environment’ (CVE) refers to a virtual world that is depicted in a computer simulation, and which is shared between two or more computers. Several frameworks such as DIVE [45], SIMNET [44] and MASSIVE [19] have been
developed to support the implementation of CVEs. They supply tools to handle data replication and transmission across the network.

Typically, the representation of this virtual world is duplicated on all collaborating systems, so that when the participating users view these representations, they feel as if they are participating in the one, common virtual environment (Fig 2). As users interact with virtual objects in the environment, the changes need to also be registered on all connected systems. To maintain logical consistency between these computer systems, the current state of each system must constantly be updated with changes from others. This is one of the major challenges in such a collaborative environment [96]. The Communication section of this paper describes the mechanism that was used to collect and transfer the model changes to avoid consistency errors.

Another challenge in collaborative systems that use real networks (i.e. those that go beyond the experimental situation found within a research laboratory), is that of dealing with latency in the network. Any network, whether using copper wire, fibre-optics or airwave transmissions, has an inherent time to deliver data from one machine to the next, due to the inflexible speed of light. However, often a greater contribution to the delay in network connections is that introduced by queuing in the routers that may be necessary in a multi-purpose network. This latency can result in the late arrival of update data. Because of the late arrival at a local machine, the local user may, by chance, interact with some part of the model that has already been changed by another user. This can lead to inconsistencies and, especially in a haptically enabled configuration, can lead to oscillations and instability, potentially rendering the system unusable [53] [145] [160]. Gunn et al. [83] describe the use of ‘pseudo physics’ to accommodate this unavoidable latency, when dealing with interconnected elastic tissue.

Fig. 13.2: Dual sculpting
simulations in a collaborative surgical simulation. With digital sculpting, such measures were found to be unnecessary due to the success of using haptics to prevent simultaneous editing of the same location (see Communication section) and the fact that principally plastic, rather than elastic deformation was required to emulate the properties of clay. The consequence of the latter point is that only the region of the surface that is in contact with the haptic tool deforms and it stops deforming as soon as the contact ceases. Also the haptic properties of the tools cause them to collide with each other and therefore prevent simultaneous touching of the same area of surface. Elastic deformation, on the other hand, could allow a surface to continue to move after the sculpting tool has left the site and also typically has surface stretching effects extending beyond the contact point. Details of the surface model used for the sculpting are described in the Program Structure section.

13.2.4 Features

The sculpting system has several modes of operation. Each user can choose to sculpt, paint and draw independently of their partner on the collaborating system. They can also specify their clay material to have a different viscosity and cohesiveness. By default, both ends of the system start with a spherical ball of virtual clay, but any VRML model can be chosen as a starting point. After the two systems achieve connectivity through the network, all subsequent changes are conveyed bi-laterally between them.

The surface-sculpting activity is carried out with one of a number of different sized spherical sculpting tools, affixed to the end of a virtual stylus. The sculptor strokes the

**FIG. 13.3:** A) Simultaneous ‘Potters Wheel’ and stationary sculpting. B) Painting and sculpting

203
tool on the surface resulting in permanent deformation around the path of the tool. This provides the ability to produce indentations into the surface, but does not provide a mechanism to pull the surface outwards. To achieve this, the tool can be ‘popped-through’ the surface by striking it quite hard. The spherical tool tip passes through the surface without leaving any indentation and is henceforth on the inside of the clay. It can then be used in the same way as when on the outside. Using a stroking motion against the inside of the surface bends the surface outwards. The tool tip location is always apparent to the user due to the virtual stylus protruding from the model. The behaviour is as if the user is sculpting a pliable bubble from either the outside or the inside. The ability to have two sculptors working together in a virtual environment has some interesting benefits. They can simultaneously be working at different zoom levels and can be working with different virtual tools. It should be noted that the volume of the clay is not conserved. This can be seen as a disadvantage since it does not reflect reality. However there are times when it provides a benefit; the sculptor can be assured that modifications to one part of the model will in no way change any other part that may already have been completed. The surface is modified only in the immediate vicinity of the haptic tool’s contact.

When switched to painting mode, the user can select a colour and apply this to the surface of the model with a simulated paint brush. In this mode the surface does not deform and is rigid to the touch. Each touch of the brush to the surface replaces whatever former colour was applied to the spot being touched. An extension to the painting mode is the ‘surface draw’ mode. In this case the successively detected touch points on the surface are connected with a surface painted line, so that the tool behaves as a ball point pen. It is implemented through modifying a texture which is mapped onto the surface, in real time. Figure 13.3b shows the screen with one user painting while the other sculpts. With each of these modes the model can be returned to sculpture mode and sculpted after it has been painted. This is an example of a capability that does not exist in the real world.

Since the virtual clay is approximated with a polygonal model, any surface deformation causes the polygons to stretch or compress. This can result in the situation where some of the polygons become large enough to become individually noticeable. Also, fine work may be needed on one part of the model, but not on other parts. The ‘paint on resolution’ feature of the system solves both of these problems. Having chosen this
mode, the user is provided with a paint brush as a tool, but instead of painting a colour onto the surface, they paint ‘sprinkles of gold dust’. This provides an indicator of the area that they have covered. When they return to sculpting mode, the system subdivides all the polygons beneath the gold dust and the local surface area quadruples its resolution. The process can be repeated up to the memory capacity of the system. It should be noted that this feature is currently only available in stand-alone (not collaborative) mode. However, there is no technical barrier to it being available across the network.

During testing we learned that users typically chose to sculpt a human face. To facilitate this, a ‘symmetrical sculpting’ mode was provided. This duplicates any deformation that the user performs on either side of the Y-Z plane. Using this feature, users can sculpt one eye and get two, one ear and get a second or create a perfectly symmetrical nose. After returning to normal sculpting, they have the option of introducing asymmetrical characteristics.

To allow the system to be used as a potter’s wheel, a ‘spin’ feature was added. The clay model will spin around one of the X, Y or Z axes. When the sculpting or painting tool is touched against the spinning surface it sculpt a groove around the model or, in paint mode, it paints a circular line. In fact the whole creative space (but not the user’s tool) is rotating, so that if the user chooses free line drawing mode along with spinning, a ‘spun’ thread in space is created. Interesting circular and spiral shapes can be produced with this combination.

As mentioned before, each of the collaborating systems has complete independence of choice of operation mode. This means that it is possible for one end to be spinning while the other is stationary. One user can be treating the model as a potter’s wheel, carving grooves or ridges, or hollowing out the interior of a bowl. Meanwhile the other user can have the model stationary, perhaps working in detail on some feature (figure 13.3a). In this configuration, the stationary user sees the spinning user’s tool revolving around the model like a satellite around the earth, perhaps carving a groove as it goes. The other user sees both the clay model and stationary user’s tool spinning around the chosen axis, as if it were a spinning globe of the earth with a person working on the surface. Collisions between the two users’ tools are discussed in the Communication section.
As well as this, it is possible for the properties of the surface to differ between the two systems. The clay could appear stiff to one sculptor and soft and pliable to the other.

### 13.2.5 Program Structure

The 3D scene is constructed in the style of VRML, with a scene-graph consisting of a hierarchy of nodes. For example there are Transform nodes, Shape nodes, Appearance nodes and Geometry nodes along with various specializations of these nodes. These nodes store their attributes in the form of fields held within the nodes. For example the Transform node has translation, rotation and scale fields as well as a field to store the children nodes to which the transform applies. The fields can be connected together by routes, which can pass messages to other fields (typically in other nodes) whenever a change to a field occurs.

The application is built upon the Reachin API (formerly Magma) [180], which provides the basic structure of node interconnection and field routing as well as graphic rendering and solid surface haptics. To create the sculpting program, selected Reachin nodes were specialized into types specific to the application, using C++ inheritance. An example of one such specialization is the PaintPot node, which is a specialization of a Group node, and which implements a virtual paint pot UI widget for choosing paint colours. The Compass node is a specialization of the Transform node and provides a visual indication of the current global orientation of the clay model.

The DeformableTriSurface node provides the main sculpting functionality. It inherits from the IndexedFaceSet node and encapsulates the sculpting and painting algorithms that make up the most interesting part of the application. It has fields which describe its pliability and others that hold the actual vertices and connectivity list that makes up the triangular polygons of the surface. The user’s tool position, routed to the DeformableTriSurface, triggers one of several possible changes in the clay model, depending on the selected mode of operation. In sculpt mode, the spherical tool location is tested against the surface vertices of the model. If any vertex is within the tool’s volume, the penetration adds to the force reflected back to the user through the haptic device. Simultaneously, the vertex is moved a proportion of the penetration distance back towards the surface of the tool. Note that it is not moved completely to the tool surface, since this would result in an almost instantaneous elimination of any vertex penetration, which would then remove any force reflection back to the user. The consequence of this would be that the user would feel no effective force at all as they
press into the clay. Neighbouring vertices are moved proportionally to their distance away from the nearest sculpted vertex. The proportion can be altered to allow for different cohesiveness values in the model.

Since the repositioning of the vertices affects the haptic feel of the surface, it needs to be updated at around 1000 Hz. Any update rate below this can result in a buzzing vibration in the haptic device. The graphic display, however, can be updated at only 20-30 Hz before the user notices any adverse effect. For this reason a separate thread, looping at 1000Hz, controls the vertex positioning algorithm and all haptics in the system. Then, every 1/30 s the accumulated vertex position changes are passed to the graphics thread for visual display. Since the haptics thread has only 1 ms to perform its calculations, there is not enough time for it to check each vertex against collisions with the sculpting tool. However, since the user cannot move the tool a great distance within the 1/30 s graphics cycle, the graphics thread can collect all vertices that are within ‘striking range’ of the sculpting tool in that time interval and pass this temporary list to the haptics thread. The smaller list is manageable in the 1 ms time available. Nonetheless, the very act of determining which vertices are within range can take significant time. To overcome this, vertex references are kept in a spatial index structure of axis aligned cells. Once we know that the sculpting tool is within a cell, it is efficient to retrieve all vertices in that cell, as well as a sufficient number of surrounding cells. It can then pass the list across to the haptics thread as candidates for collision. The haptics thread then needs to only scan this substantially smaller list for collisions. Naturally, as vertices are moved by the sculpting action, they often end up in another spatial index cell. Since the cells are axis aligned, the new cell can be easily calculated from the new vertex position, and the vertex is re-allocated to the new cell. The penalty of this calculation is more than compensated for by the resulting saving in search time for collisions.

In paint mode, the surface of the clay reacts to being touched with the painting tool tip. It then calculates the 2D spot on the texture that is currently being mapped onto the surface. It changes a number of pixels, (depending on the paint brush size) to the current paint colour. The surface draw mode works the same way except that it calculates a contiguous line of pixels between successive contact points until the brush is lifted from the surface significantly. The algorithm, therefore, accommodates small involuntary bounces in the virtual pen.
In free line draw mode, the user can draw lines in free space, not necessarily on the surface of the model. Although the lines appear to be equivalent to those drawn on the surface, they have a different underlying structure and behaviour. A ‘free line’ consists of a DeformableLine node which inherits from the IndexedLineSet node of the Reachin API. As the user drags the haptic tool with its button pressed, successive vertices are added to the line until the button is released. When the user is sculpting, the line vertices are handled in the same way as for the clay surface, reacting to collisions with the tool and deforming accordingly, by being repositioned a percentage of their penetration distance inside the tool. A list of all deformable objects (both lines and surfaces) is maintained by the system, and this is visited once on each update cycle. Each object is checked for collisions and treated independently, thus allowing one movement of the haptic tool to sculpt all colliding surfaces and lines simultaneously. There is no mode shifting required between sculpting haptic lines and surfaces.

13.2.6 Communications
The inter-program communication occurs through sockets, using both TCP/IP, and UDP/IP. Reading and writing to the network occurs through separate, dedicated threads.

We use replicated databases on each machine with update data being transferred between them when necessary. We have extended the field-route concept to ‘remote routes’ that exist across a network connection. There is a specialized node in the scene-graph that handles the network connection and holds the remote routes. When local fields are connected to these remote routes, any change in the local field is transmitted to the other end of the network connection. If the receiving end has the remote route connected to a field, that field gets a change-message in the same way that it would if it was receiving it from a local route. This mechanism provides an easy API to facilitate the connection of selected components of a scene-graph, together across a network. In the sculpting application, the vertices of the clay surface are held in a field of a node, which is routed to one of the remote routes. When some vertices are changed by the user’s sculpting actions, the new vertex positions and their indices are transmitted, via the remote route, to the far end and its vertex field is subsequently updated. Some of the other fields that are transmitted are tool position, tool orientation, paint spot size and paint spot location.

Networks have an inherent latency in the delivery of data from one location to another. For example, we have found that a typical latency between Australia and the USA is
about 190 ms. The network distance, network quality, bandwidth and the number of router queues that the data must pass through, all have a bearing on the degree of this latency. With such a delay in the delivery of messages, it is possible that one user can move a vertex in the model at the same time that the other user is changing it in another direction. The two users do not see the current state at the other end, only a state that existed some time in the past. Their immediate actions may be based on a state that is no longer valid. The system needs a mechanism to determine how to resolve such inconsistencies.

Parallel modifications at the same location require very precise synchronization. Both network latency and the high update rates needed by haptic feedback exacerbate the problem. When the requirement for haptic refresh cycle times of around 1 ms are combined with network latencies of up to 190 ms, current technology does not allow independent simultaneous editing of a haptic surface, without the likelihood of introducing physical feedback and instability. One solution to this is to allow the users’ actions to be combined into a resultant action before being applied to the model. Such a method is used in [83] where one of the collaborating machines is nominated as a physics server and all user inputs, from both ends of the network, are passed to that machine. It then determines the resultant forces on objects and how they should move, and informs both ends of the new object positions for separate rendering. Another method is to have a locking mechanism so that one end ‘claims’ part of the model for editing and has ownership of that part until released. The other end cannot edit that part for the same period [129] [130]. This may require the user to be aware of parts of the model that are temporarily unavailable for editing, and their editing behaviour must be modified accordingly.

In this application, it was possible to avoid introducing software complexity in the underlying system, and any special behaviours on the part of the users, due to the fact that there was a haptic system available. In the real world, two sculptors would not sculpt the same point on the piece of clay at the same time, since their hands or sculpting tools would collide, preventing them doing so. As mentioned in the introduction, a virtual environment allows sculptors to work without the inconvenience of collisions between their physical bodies, delivering advantages in freedom of movement and access. Having achieved this, we then have the option of selectively reintroducing inter-personal collisions where there may be a greater advantage in doing
so. Such an opportunity presents itself in addressing the problem of mutual simultaneous editing. By enabling haptic collision between only the virtual tool tips, we can prevent simultaneous vertex manipulation by physical exclusion from the zone closely around the sculpting point. Since it is only the area within the radius of the tool tips that can collide, this still essentially permits sculptors to ‘pass through’ each other, as mentioned earlier, with the exception of the tool tip. This is functionally different to sculptors’ bodies colliding, since the collision of bodies serves no purpose towards the outcome, whereas colliding tool tips serves the purpose of preventing simultaneous editing of the same point. I considered this compromise a worthwhile trade-off from the original aim of unhindered access, since it solves the simultaneous editing problem without requiring any contrived behaviour, or knowledge of the underlying system, on the part of the participants.

In practice, the haptic contact between the tool tips involves more than just supplying the tool tips with a haptic, ‘touchable’ surface. The tool position is transmitted with TCP/IP [220] to the other end at the graphics refresh rate, 30 Hz. At this rate, a hard-surfac ed tool tip would move slightly, each 30th of a second. This causes an uncomfortable vibration when the local user touches it. Even sending the tool tip position at higher rates can cause a higher frequency vibration, due to the latency in the network and correspondingly delayed reactions to any movement. The problem is solved by using an implicit function to represent the tool tip, allowing the haptic influence to extend slightly beyond the tool surface and introducing some damping into the inter-tool interaction. The minimum network transmission rate for the tool position that could achieve stability over a network with up to 300MSec latency was found to be 400 Hz. It was also necessary to use UDP/IP, ibid, for the transmission protocol. UDP, although less reliable, has no buffering delays and has lower jitter due to the fact that it does not try to resend lost packets. In our case, an occasional lost packet is preferable to the delay introduced by checking for them.

With other data, such as the segments of a drawn line, it is important that each position that is sent is actually received and displayed, so TCP was used. Another problem that can occur is that if the receiving machine is running at a slower frame-rate than the sender, it is possible for data to be sent faster than the receiving machine can read and process them. In this case, some values may be lost, resulting in an incorrect rendering
of the line. A buffering mechanism was introduced to avoid this by ensuring that all received values are recorded and applied as soon as the system can accommodate them.

The collaborative sculpting system incorporates a shared virtual 3D scene in which users can see each others actions, tool types and changes in the model as they occur. It also has an audio and video link to allow users to discuss their work. Using these features, a co-operative work situation is created which allows them to use language, gesture and demonstration to illustrate their ideas and thoughts. While it is still possible for the sculptors to work independently in parallel, the system encourages the sharing of ideas through the communication channels provided. An illustrative example was captured in the conversation of two subjects in the trial detailed later in the Creativity Experiment section.

Subject 1: “What’s that? I’m doing the front”

Subject 2: “No, I’m doing the face. Look, here’s the nose”

Subject 1: “But what’s that spiky stuff then?”

Because of the shared 3D environment and the awareness of each other’s tools, the users can use diexis, as exemplified by the use of the words “that”, “here’s” which were accompanied by pointing and gesture. Similarly, it is possible for one sculptor to demonstrate a shape or style by example using the clay itself.

We found that it was possible to run collaborative sculpting sessions over distances across Australia and also between Australia and the USA. They were connected via the CeNTIE network in Australia [48] and Internet2 [106] in the USA, and used the AARNET [1] link using the Souther Cross Cable Network [217] across the Pacific Ocean. This had a network latency of around 190 ms. In laboratory tests the system worked satisfactorily with simulated latencies of up to 300 ms. The clay model involved 22,528 triangular polygons. This resulted in a maximum data transfer of 1 MBits/second during deep sculpting (discounting the separate video / audio link).

13.2.7 User Interface

The user interface comprises a series of haptic virtual buttons within the 3D scene, a compass widget to indicate the current orientation and a rack of paint pots for choosing colours. One interesting discovery was that in UI design for virtual environments, one should not always strive for maximum realism. Initially I created the paint pots with a
hard, ‘glass’ surface, and soft, sticky paint inside (with a dribble of paint running down
the outside). The idea was that users would dip the brush into the paint to select a
colour. In practice, however, we found that users typically tried to pass the brush
through the side of the paint pots. This could be because they were all computer users of
some sort, and have come to expect ‘magic’ behaviour when they know a computer is
creating their work environment. The solution to the problem was to make the glass jar
react to the touch of a brush, as well as its contents.

The mode of operation is reflected by the graphical style of the virtual tool in the scene.
The remote user can see both their own tool, and the remote user’s tool, so they know
what mode they have chosen and what mode the remote user is in. There was therefore
no need to reflect the state of the each user’s buttons and widgets at the other end.

13.2.8 Creativity Experiment

Can real-time collaborative sculpting increase the creativity of the individual artists? It
is hard to imagine any quantitative experiment that could measure this. However, it is
possible to get some qualitative idea on whether the ability to share the sculpting
experience with another person, helps with inspiration. A small experiment, using 16
subjects, was devised to try to determine if the shared workspace enhanced the
participants’ creativity. They were individually given an introduction to the sculpting
system and were then paired up with a partner on a networked, collaborating sculpting
station. They were then allowed to sculpt a spherical ball of virtual clay, together, for

![Creativity Experiment Results]

Fig. 13.4: Creativity experiment results. a) Number of subjects with each score
b) Subjects’ scores, grouped in pairs.
about ten minutes (actually, often they wanted to keep going).

Nearing the end of the allotted time, they were asked to grade the amount that their sculpting ideas were triggered by something their partner had done. A scale of 1 - 10 was used, where 1 indicated that they felt that they had no ideas triggered by their partner – i.e. they were effectively working independently. A score of 10 implied that they were totally reacting to what the other person was doing. Scores of 2 - 9 indicated degrees of influence between those extremes. It was stressed that the point to note was how much the other person’s actions triggered a creative idea of their own that they then pursued, not just how much their sculpting actions were controlled by those of the other.

There were 6 females and 10 males between the ages of 12 and 53, with varying experience of computer interfaces. None professed to have particular artistic abilities. Figure 13.4a shows the number of subjects who nominated each possible score. The average nominated score was 4.43, with a standard deviation of 1.89, indicating that a shared environment can indeed enhance an individual’s creativity.

Figure 13.4b shows the actual scores nominated by the subjects, grouped into their pairings. There appears to be no obvious correlation between the score difference of each pair and their overall score, i.e. that pairs necessarily agreed on the sharing experience to any degree, or that pairs nominating a high score agreed any more than those nominating a low score.

Since a score of around 1 would have indicated that no new ideas were generated by the presence of the partner, the mean of 4.43 indicates that a number of ideas were typically generated. Figure 13.4b shows that both sculptors in each pair typically recorded the generation of new ideas from their partner’s ongoing work, suggesting a two-way flow of increased creativity. This creativity flow could have only been in a one-way direction if the sculpting had occurred consecutively rather than in parallel.

The subjects for this experiment were chosen from a general cross-section of the population. A further trial was undertaken to investigate whether the collaborative system shows potential in the eyes of experienced artists. Two instructors and two students from the Australian National University School of Arts, Ceramics Workshop were given the opportunity to use the system and their comments and suggestions were noted. They felt that the collaborative simultaneous working environment was fairly chaotic to start with, but put that down to the novelty of the concept and inexperience
with the hardware. One comment was that the immediacy of the collaborative interaction resulted in a somewhat *somatic* as opposed to *intellectual* response to the partner’s work. Their experience of collaborative work had previously been sequential, allowing the second artist time to intellectualise their response to the first artist’s creation. They thought that an immediate, somatic response was very interesting and held potential for a different style of creativity. Most preferred to divide the task so that one artist worked on form while the other worked on surface. One student commented that she noticed a swirl that the other was creating, which seemed to have the appearance of the top of a shell, so she started hollowing out the inside to match. Some of the creations are shown in figure 13.5.

13.2.9 Future Work

The application currently provides each sculptor with only one sculpting tool. The author has been involved in the development of haptic medical systems which allow two collaborating surgical trainees to use two haptic tools each, totalling four haptic devices at once in the system [102]. The virtual model can recognise contact from each
tool independently and can therefore behave differently depending on the tool type. It is planned to import this mechanism to the sculpting application to allow two-handed interaction.

The same medical haptic application allows the drilling and boring of simulated human bone. This technique could also be introduced to the sculpting application to provide a range of materials from soft pliable clay to hard stone.

13.2.10 Conclusion

This application showed that it is possible for two sculptors to simultaneous work on a virtual clay model in any mode they chose without any interference from the other user. Users’ actions could be completely asynchronous, as they would be in the real world. They could work individually or co-operatively, and be in the same, or different, co-ordinate systems while separated by thousands of kilometres.

Our qualitative study suggests that the system can indeed improve the creative output of the two sculptors. We can also look at a parallel situation to draw some subjective conclusions about the effect of a collaborative environment on creativity. Pair programming is a form of writing computer code where two or more programmers look at a shared display. They make suggestions to each other and add to the logic as they progress. It has been shown that the act of sharing the production of the computer code improves the resulting product [247]. One explanation of this effect is that as one programmer adds a piece of logic, it triggers some lateral thought in the other, which can result in more alternatives being explored. It seems logical that the same effect would occur with two artists working simultaneously on a sculpture. As one artist changes the form of a part of the model, it may suggest new ideas and forms to the other. This process could play back and forth between two artists until the work is completed.

Traditional sculpture has been carried out by a single artist, or two or more artists in a sequential mode, or in the apprenticeship model with a chief artist creating the major part and several assistants filling in minor details. This virtual environment allows all of these modes of operation as well allowing two artists to work closely together equally without getting in each other’s way. By increasing the number of modes of creation available to the artists, it is likely that their overall creative output would be enhanced.
13.3 Discussion

This paper investigated the correct balance between the provision of sufficient realism within a virtual environment to induce a “suspension of disbelief” and simultaneously supplying unreal qualities that, together, can stimulate an experience beyond that which is achievable in the real world. In particular, it is presenting a shared, touchable, 3D view of a clay model, but a model which the users can choose to simultaneously experience from within different co-ordinate systems, and without the constraints of interacting with that ‘world’, from within their physical bodies. The results show that such a system does indeed provide an element of creativity that might otherwise not have occurred. The fact that the respondents in the user study reported that they gained ideas from the actions of their partner is evidence of this.

13.3.1 Improving the Model Resolution

The processing power of the computer limits the number of polygons that can comprise the virtual clay model, thus putting an upper bound on the resolution and smoothness of the surface. The source of this constraint is the search time for colliding triangle vertices. As mentioned, one technique for extending this limit is to construct a spatial index of cells, with each cell containing a list of its spatially contained vertices. Whenever a vertex is moved, as a result of the sculpting process, it may need to be reassigned to a new cell. The cells are stored in a linear array, indexed by a cell index. The algorithm to calculate the cell index for a vertex is shown in equation 13.1.
During the collision detection algorithm, the tool position is used to find the cell in which it is contained, also according to equation 13.1. Since cells are structures which hold a list of contained vertices, the vertices from this cell are candidates for collision with the tool. However, the vertices in the surrounding cells may also collide with the tool during the next cycle. These surrounding cells are retrieved by applying a number of pre-calculated offsets to the original cell index, giving a roughly spherical selection of surrounding cells. These pre-computed offsets are stored in a list of their own and used repeatedly, from different centre-cells, as the tool moves. The offsets are calculated at the program start-up or whenever the tool size is changed, according to algorithm 13.2.

\[
\begin{align*}
    a &= \frac{V_x - \min_x}{\text{incr}_x} \\
    b &= \frac{V_y - \min_y}{\text{incr}_y} \\
    c &= \frac{V_z - \min_z}{\text{incr}_z} \\
    \text{cell\_index} &= c \times n_x \times n_y + b \times n_x + a \\
\end{align*}
\]

where \( V_x \) = vertex \( n_x \) = number of cells (in \( x, y, z \))

\( \text{incr}_x \) = x size of a spatial index cell
\( \text{incr}_y \) = y size of a spatial index cell
\( \text{incr}_z \) = z size of a spatial index cell
\( \min_x \) = minimum \( x \) value of spatial index
\( \min_y \) = minimum \( y \) value of spatial index
\( \min_z \) = minimum \( z \) value of spatial index

(13.1)
This sculpting system is also limited by the fact that the sculpting occurs by stretching surface polygons. As mentioned in the paper, these polygons can optionally be subdivided by the ‘paint-on-resolution’ tool. However, this is a separate task that a user needs to undertake when they are noticing a loss of precision in their work. By the time they have noticed this, it is likely that there may already be some areas that they need to go back and repair once the resolution is high enough to do so. A better solution might be to subdivide triangles on-the-fly; whenever a triangle is stretched beyond a certain point. The stretching could be detected by the system and the triangle’s edges would be automatically bisected producing four new triangles which would replace the original. This has not yet been implemented.

13.3.2 Whole Handed Sculpting

Users of the system have commented that they would like to be able to use their whole hands instead of holding a single instrument. Haptic glove systems do exist [184] [105], but currently they take considerable time to calibrate to a particular person’s hand and they have a resolution and response rate much lower than that of the single point haptic device. With improvements in the technology plus increasing speed of computer processors, such systems may achieve feasible whole handed virtual sculpting.

13.3.3 Surface Problems

One unresolved problem with the system stems from the fact that a surface is being sculpted, not a solid voxelated model. While this has advantages in requiring less computation, and permitting sculpting from inside the model as well as outside, it can
result in pieces of surface folding over on top of each other. Once this has happened, it is virtually impossible for the sculptor to correct the situation, as the sculpting tool cannot get into the small crack between the surfaces. Worse than this, there is currently no inter-surface collision detection – i.e. the surface vertices are not checked for collision between themselves – only against the sculpting tool. This means that it is possible for a surface to fold back and through itself. These problems are as-yet, unresolved, but fortunately they do not occur frequently. One possible solution might be to have an effective repulsion between vertex nodes. If any node is pushed closer than a certain minimum limit to another node, it ‘springs back’ to that limit. This has not yet been implemented.

13.3.4 Data Transmission

The clay model comprises triangles connecting vertices which are held in an array structure within a deformable IndexedFaceSet node in the scene-graph. As the sculpting tool interacts with the surface, a number of vertices move and their new positions must be transmitted across the network to the collaborating machine. However, the format of the vertex data is fundamentally different to the format of the data types transmitted in the gall bladder surgery discussed in earlier chapters (Chapter 9, Chapter 10 and Chapter 11). The voxel data is stored in an array within one VRML data object (a derivation of an IndexedFaceSet). The standard mechanism employed by the communications software framework (see section 9.2.9) is for a RemoteField object of one of the particular types (such as RemoteSFVec3f) to be notified via a route when the data object changes, and for that RemoteField to handle transmission of the changes to the remote machine. However, in the case of an array, this mechanism does not provide any information on exactly which elements of the array have changed; only that something in the array has changed. Continuing with such a system would require sending all elements of the array each time any (typically small) number of the elements changed, thus introducing an unacceptably large amount of network traffic.

The solution to the problem required the introduction of a separate mechanism for array data types, encapsulated in a new remote field class, RemoteNodeField, which can send updates of partial contents of a given array field of a node across a network. It differs from the single field classes (such as RemoteSFVec3f) in that it does not remotely route a field in one node directly to a field in another node, but instead holds a reference to a particular node that contains the field, as well as a reference to the field within that
node. To trigger the mechanism, the sculpting code must maintain a separate array field that holds the *indices of the vertices* that have just been changed. It is this multiple indices field which is routed to the *RemoteNodeField*, not the vertex field as before. Once the RemoteNodeField is notified that a change has occurred (through the field-route mechanism) it reads the new indices, retrieves only those vertices from the node held in its reference, and builds a packet of data holding both sets of data for transmission on the network. This is shown diagrammatically in figure 13.6.

The corresponding RemoteNodeField on the receiving machine reads these indices and values from the network and changes them in its own referenced node.
Chapter 14.

Conclusion

Humans live in an environment in which touch is an every-day, integral part of their existence. Even those unfortunate enough to have impaired senses of sight, hearing and even tactile sensation, can experience the resistance that real world objects provide, when these come into contact with their bodies. The human race has evolved over millions of years to associate the intersection of their motion, relative to the objects in the world around them, with the experience of a force on their body. From the unpleasant strike of a toe onto a rock to the subtle touch of a leaf brushing the face, haptic sensation is around us all of the time. The combination of sight with touch allows us to grasp, hold, manipulate, move and inspect the objects that surround us in our daily lives.

With this fundamental component of our existence, it is surprising that the norm in the computing world is an environment devoid of touch. Objects that our computers present to us typically take on a visual form but have no physical interaction qualities; we do not feel them and if we close our eyes we are unaware of their existence. This is so foreign to our experience in the natural world that it is a tribute to human adaptability that we can indeed operate at all in such an environment.

The vast majority of computer work is carried on in a two dimensional space; either with lines of text or ‘windows’ representing paper documents on a flat screen. These windows can pass through each other (coming to the top of a virtual stack) at the click of a button, without having to be slid out and around the edges of the stack, as would be the case with real sheets of paper. This is very convenient, once the concept is understood. The world of computing, and windowing user-interfaces in particular, often
violates the laws of physics for convenience; our human species may have evolved to operate in a particular physical environment, but that is not to say that every aspect of that environment is always convenient for us. When we have the ability to construct a different reality (a virtual reality) we are also free to pick and choose which aspects of the real world to reproduce, which to leave out and which to modify, to attain a compatible blend to suit our purposes.

Three dimensional motions are more complex than those in two dimensions. The motion of a human hand in space is the result of several individual, but co-ordinated, angular movements of the wrist, elbow and shoulder joints [143]. In fact, in some cases there can be more than one different combination of joint angular motion that produces the same hand motion [58]. When using a computer mouse, the table upon which it rests provides a constraining haptic force, thus simplifying the actions required to control it. Three dimensional interaction in the computer environment has no convenient aid such as this, unless it is artificially added using haptic feedback. Imagine a mime artist acting out the cleaning of a window. His hand actions require great skill to present a convincing representation, whereas to do the same task with a real window is trivial, simply because the haptic resistance of the window makes it easy to stay on the surface.

3D interaction with virtual objects within the computer environment can be similarly aided by introducing a haptically induced touchable surface. Chapter 3 describes an application that allows sculptors to shape, paint and draw using the benefit of touchable surfaces on their artworks. When the artist is painting, the same software mechanism that provides the haptic feedback at the moment of touch is employed to trigger the simulation of paint application at that same contact point. When sculpting, the amount of movement of the clay surface is used to determine the amount of resistive force that the sculptor feels. That is to say the modification that the sculptor performs on the environment causes the force that the sculptor feels in the same way as occurs in the real world. The application is simulating the real world in the aspects that are convenient for the sculptor.

However, not everything about the real world may be convenient to the sculptor. Chapter 13 investigates the introduction of behaviour that is unrealistic with respect to the real world, such as allowing the model to spin for one sculptor but be simultaneously stationary for another. Also the ability to pass the sculpting tool through the model surface to permit sculpting from the inside out has no parallel in the real
world. These aspects of the application were found to be interesting features by the ceramics artists involved in its trial.

The utility of providing realistic touchable surfaces for virtual clay sculpting was researched in the first experiment of the user study reported in Chapter 7. It was found that skill at sculpting a smooth surface could be enhanced by providing a haptic surface to the clay. It was also found that this improvement was more marked in sculptors with generally lower overall skill at the task, indicating that haptic feedback could be particularly valuable for users with lower dextrous ability. Such a result may come into play when faced with a decision of whether to incur the expense of adding haptic feedback to an application. If users are expected to be novices or unskilled at the task, haptic feedback should be of considerable assistance. If they are expected to be skilled at the task, the cost-benefit trade-off may not be sufficient to warrant the added expense.

The work described in this thesis has shown that the provision of haptic capabilities within a virtual environment can assist a user in a task. It also provides a description of a functional hardware configuration and a software framework for the development of such applications. (The framework described in this thesis refers to development based on the Reachin API, but could equally well be applied to other commercially available haptic/graphic APIs 14). It consists of firstly building any nodes to perform any specialised functions that do not exist in the API. This is discussed in sections 3.2.6, 3.2.7, 3.3.10, 12.3.7 and 13.3.4 and examples of the force calculation algorithms that can be contained within these nodes are covered in sections 5.3.1, 6.3.2 and 9.2.8. The next step involves assembling these into a scene-graph using Python [177] scripts and/or VRML files and making the necessary field connections as discussed in sections 9.2.9 and 13.2.5. The ReachinLoad process is then run, to read these files and start the application. If used along with the Reachin API documentation, the thesis provides a proven method of developing immersive, interactive haptic programs for a wide variety of applications.

A large part of the thesis covers the issues surrounding the networking of haptic applications together. At about the same time that the first publication of this component of the work appeared (“Trans World Haptics”, Chapter 8), another successful long

14 The ReachinAPI was used for this work. Other APIs include H3D and OpenHaptics.
distance haptic interaction was reported in the media. It involved a user on each side of the Atlantic Ocean simultaneously pushing a blue cube from opposite sides. The network distance for this was approximately 5,000 km. By contrast, the networked haptics work reported in this thesis involved a complex set of deformable objects, interconnected with simulated elastic tissue (the body organs), and operating across a network distance of 22,000 km (Canberra to Stockholm). This work claimed the world record for long distance haptic collaboration in a virtual environment and to our knowledge, the record still stands. The thesis has shown that, contrary to other reports, simultaneous manipulation of elastic and plastic objects by users located on opposite sides of the globe is possible, and it provides a detailed description of the mechanisms that can be employed to successfully overcome the inherent instability introduced by the network latency (Chapter 9).

The thesis describes the application of these haptics techniques to two surgical training scenarios (Chapters 10 and 12) and discusses a method of extending this capability to a classroom environment involving groups of students. A clinical trial of the mentoring effectiveness is reported by Chapter 12, and shows that students felt that the system was a valuable tool for learning the surgical skills required for the delicate procedure of temporal bone drilling. It is likely that similar techniques could be used for dental, cranio-facial and orthopaedic training.

Although the networking capability was developed with long distance teaching and mentoring in mind, it is interesting to note that the greater benefit of the networking capability has proven to be in situations where both networked machines are located side-by-side in the same room. Without the ability to combine two separate users in the one virtual environment, an instructor has no way of pointing and sketching within the view that the student is operating. Using the hardware described (sections 3.2.2, 5.2.2), but without networking, the instructor cannot physically point to objects within the image in the mirror as the hand occludes the 3D scene and also has no depth location, and s/he cannot reach under the mirror to point where the objects actually appear, as the hand is then hidden. With the networked environment, the instructor can not only gesture, point and sketch in amongst the 3D objects, but s/he can also grasp and move those objects in collaboration with the student and even grasp and guide the student’s hand. Although the ability to do this over global distances exists, it is more likely that the networking feature will be most commonly used in the side-by-side configuration.
This was a pleasant and unexpected result and adds greatly to the utility of the networking configuration. As was discussed in Chapter 11, it is possible for others to follow the instructor-student interaction, also in 3D, and a combination of an instructor, one or more interacting students and several observers is likely to become an effective teaching design for 3D skills and anatomical investigation.

Medical colleagues have suggested that this networked virtual environment could also be functional in a surgical planning phase, allowing inspection and discussion amongst surgeons prior to a surgical procedure. To realize this outcome, it would be necessary to build the 3D anatomical model from medical scans specific to the patient in question, a process which is not currently possible with automatic techniques. However, should this become feasible and economic, the haptic system described here is likely to provide a suitable user interface environment in which to carry on that discussion.

The creativity study on collaborative sculpting, covered in Chapter 13, demonstrates that this technology is by no means limited to the medical domain. Any environment that involves three dimensional work, and especially co-operative work with others, could be a candidate for implementation using the hardware and software techniques covered by this thesis. Fields such as theatrical design, choreography, factory design, architecture, film direction, town planning, landscaping, assembly planning, product design and various forms of engineering all fall into this category. The work covered by this thesis has shown that haptic feedback can assist a user in a 3D task and that networking allows collaborative editing and manipulation of objects with a variety of differing physical properties. Three dimensional editing environments could be tailored to apply to these domains using the techniques investigated here.

I conclude that the thesis statement:

Artificial force feedback can enhance a user’s skill in a virtual environment and that such haptic applications can be developed, which allow users to touch, interact and generally collaborate together over distances spanning the world.

has therefore been demonstrated.
Reference List


[10] Atkinson, A.C. "Transformations for percentages and proportions". In Atkinson, A. C. "Plots, Transformations and Regression; An introduction to graphical methods of


[78] Goncharenko, I., Svinin, M., Matsumoto, S., Masui, Y., Kanou, Y., HosoeS. "Cooperative Control with Haptic Visualization in Shared Virtual Environments". In Proceedings of Eighth International Conference on Information Visualisation (IV'04),


[89] Hajian, A.Z., Howe, R.D. "Identification of the mechanical impedance at the human


[133] Lin, M.C., Baxter, W., Foskey, M., Otaduy, M.A., Scheib, V. "Haptic Interaction for Creative Processes with Simulated Media". In Proceedings of IEEE Conference on


[189] Rosenberg, L., Brave, S. "Using Force Feedback to Enhance Human Performance in


6 May 2006.


[226] Takemoto, A., Yano, K., Terashima, K. "Obstacle Avoidance Control System of Rotary


[236] van Essen, H.A., Rovers, A.F. "Layered protocols approach to analyze haptic


Appendix A :

Experiment Consent and Information Form

Information and Consent Form for participation in experiments associated with the research being conducted by Chris Gunn, a University of Western Australia PhD student.

**Purpose:** The purpose of the experiments is to assess the assistance that a robotic force feedback interface can provide in performing computer based activities. Participants will use the robotic device instead of the usual computer mouse for interaction with a computer. Participants will also be requested to view the computer screen through 3D stereo shutter glasses. The trial is expected to take approximately 3/4 an hour. The results will be incorporated into Chris Gunn’s PhD thesis.

**Data:** The movements of the robotic device will be automatically recorded and the data will be accumulated across all trials to provide a data base for statistical analysis. Participation is anonymous in that, after the trial, the data for each participant will be recorded against a number only. Participants may also be asked to fill in a questionnaire. All questions are optional. Upon completion of the research, the data will be held by Chris Gunn, and may be used for further research, however, as mentioned before there will be no identifying data associated with this.

Participants have been selected to represent a cross section of the normal population and need have no special skills.

**Risk:** There is no foreseen risk to the participant, over any risk associated with operating a standard computer. The robotic device is of low power and is many times weaker than the human arm. You will also be asked to wear 3D shutter glasses. However, before commencing the experiments, please ask any questions or mention any concerns you may have.

**Declaration:** I have read the information provided and any questions I have asked have been answered to my satisfaction. I agree to participate in this activity, realizing that I
may withdraw at any time without reason and without prejudice. I understand that all information provided is treated as strictly confidential and will not be released by the investigator unless required to by law. I have been advised as to what data is being collected, what the purpose is, and what will be done with the data upon completion of the research. I agree that research data gathered for the study may be published provided my name or other identifying information is not used.

__________________________  _______________________
Participant                                 Date

The Human Research Ethics Committee at the University of Western Australia requires that all participants are informed that, if they have any complaint regarding the manner, in which a research project is conducted, it may be given to the researcher or, alternatively to the Secretary, Human Research Ethics Committee, Registrar’s Office, University of Western Australia, 35 Stirling Highway, Crawley, WA 6009 (telephone number 6488-3703). All study participants will be provided with a copy of the Information Sheet and Consent Form for their personal records.
## Appendix B

### Sculpting Experiment Raw Results

<table>
<thead>
<tr>
<th>Subject</th>
<th>With force</th>
<th>Without force</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.4762</td>
<td>7.5027</td>
</tr>
<tr>
<td>2</td>
<td>3.0634</td>
<td>2.7412</td>
</tr>
<tr>
<td>3</td>
<td>1.9855</td>
<td>2.8084</td>
</tr>
<tr>
<td>4</td>
<td>0.4683</td>
<td>0.8622</td>
</tr>
<tr>
<td>5</td>
<td>3.5026</td>
<td>5.5166</td>
</tr>
<tr>
<td>6</td>
<td>0.3928</td>
<td>0.4499</td>
</tr>
<tr>
<td>7</td>
<td>2.0703</td>
<td>0.9986</td>
</tr>
<tr>
<td>8</td>
<td>0.4756</td>
<td>2.3614</td>
</tr>
<tr>
<td>9</td>
<td>2.9978</td>
<td>2.0192</td>
</tr>
<tr>
<td>10</td>
<td>1.9610</td>
<td>5.5888</td>
</tr>
<tr>
<td>11</td>
<td>0.7470</td>
<td>0.6380</td>
</tr>
<tr>
<td>12</td>
<td>2.9367</td>
<td>2.9830</td>
</tr>
<tr>
<td>13</td>
<td>1.0552</td>
<td>1.3816</td>
</tr>
<tr>
<td>14</td>
<td>1.4048</td>
<td>0.7560</td>
</tr>
<tr>
<td>15</td>
<td>0.9845</td>
<td>0.8799</td>
</tr>
<tr>
<td>16</td>
<td>0.7219</td>
<td>0.4016</td>
</tr>
<tr>
<td>17</td>
<td>1.5063</td>
<td>2.4162</td>
</tr>
<tr>
<td>18</td>
<td>0.6841</td>
<td>0.6437</td>
</tr>
<tr>
<td>19</td>
<td>0.4951</td>
<td>0.6249</td>
</tr>
<tr>
<td>20</td>
<td>7.3340</td>
<td>9.4387</td>
</tr>
<tr>
<td>21</td>
<td>1.9441</td>
<td>5.3075</td>
</tr>
<tr>
<td>22</td>
<td>0.3160</td>
<td>0.3831</td>
</tr>
<tr>
<td>23</td>
<td>0.7982</td>
<td>1.8030</td>
</tr>
<tr>
<td>24</td>
<td>0.7936</td>
<td>1.5846</td>
</tr>
</tbody>
</table>
Appendix C: Mine Planning Experiment Plan

1. Explain to user:

"We are going to run a series of trials where the computing system uses various different methods of warning a user about errors in what they are doing. The aim is to discover the best way of giving continuous feedback to a user about their performance when working in a 3D environment. The types of feedback are:

- Pop up message
- Red arrows in the scene
- Force feedback

You will be viewing a 3D scene through 3D shutter glasses, similar to those used in 3D movies. You will also be interacting with objects in the scene with a 3D type of mouse, called a Phantom. (Show the hardware). At times you will feel the PHANToM pushing on your hand. Before commencing the trials, you can try out the system on a few simple games, to get used to it."

2. Run snuffbox demo

3. Run puzzle demo

4. Explain how end point of pen interacts haptically but shaft doesn't.

5. Run random generator to determine the order of feedback modes.

6. For each mode, explain to the user:

Popup message:

"In this mode a popup message will appear when you are encroaching into a dangerous area. It will be accompanied by the usual Windows XP 'ding' sound for errors."

252
Arrows:

"In this mode a red line will grow and shrink between your drawing position and the danger point"

Forces:

"In this mode you will feel a force repelling you away from the danger point"

Popup message plus arrows:

"In this mode a popup message will appear when you are encroaching into a dangerous area. It will be accompanied by the usual Windows XP 'ding' sound for errors. Also a red line will grow and shrink between your drawing position and the danger point."

Forces plus arrows plus popup:

"In this mode you will feel a force repelling you away from the danger point. Also a red line will grow and shrink between your drawing position and the danger point and a popup message will appear when you are encroaching into a dangerous area."

7. Run MinePlanning application version 0 (run <ID number>)

8. Explain to user:

"In this trial you are designing an underground mining tunnel, trying to make it as short as possible, but also trying to avoid some dangerous, unstable areas. You will be presented with a 3D scene showing some fictitious geology, where rock types are represented with different colours. Rocks are semi-transparent so that different shapes can be seen behind each other. The scene is cluttered with these coloured shapes, but there is no need to understand the geology they represent. The purpose of the geology is to provide a visually cluttered, complex 3D scene that is hard to understand.

Within the scene you will see two yellow cubes. The left hand one is the start point. The right hand one is the end point. Your goal is to draw the shortest possible line between the left hand cube and the right hand cube. However, you have a second aim of avoiding some 'dangerous' areas. The centres of these dangerous areas are located at the small yellow dots, spread within the scene. Look carefully, there
should be 9 danger zones altogether. Each danger zone radiates from the central dot for a variable distance. The idea is to draw a line between the start and finish cube without going into too much danger.

You will be scored on your performance by how short your line is and how much danger you venture into.

You will repeat the trial 5 times. Each time there will be a different warning system, notifying you that you are drifting into a danger zone."

9. Select Latin Squares test mode 1, run test, record score
10. Select Latin Squares test mode 2, run test, record score
11. Select Latin Squares test mode 3, run test, record score
12. Select Latin Squares test mode 4, run test, record score
13. Select Latin Squares test mode 5, run test, record score
14. Run MinePlanning application version 1 (run1 <person's name>)
15. Explain to user:

   "Now we will repeat the test with a slight change. This time, as a mining expert, you have decided that one of the danger zones that the computer is warning you about can be completely ignored. This particular zone now has a RED centre point. The computer will still warn you about it with the various styles, but you should try to ignore that warning, while still heeding warnings from other danger zones.

   Your score will now be better if you pass straight through the red danger zone, but still be penalised for venturing into the other ones."

Identify the red danger point and repeat the trials

16. Select Latin Squares test mode 1, run test, record score
17. Select Latin Squares test mode 2, run test, record score
18. Select Latin Squares test mode 3, run test, record score
19. Select Latin Squares test mode 4, run test, record score
20. Select Latin Squares test mode 5, run test, record score.
Appendix D:

Mine Planning Experiment Raw Results

<table>
<thead>
<tr>
<th>Subject</th>
<th>Popup</th>
<th>Popup + Arrows</th>
<th>Forces</th>
<th>Arrows</th>
<th>Popup + Arrows + Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.208809</td>
<td>0.222356</td>
<td>0.104027</td>
<td>0.317479</td>
<td>0.276122</td>
</tr>
<tr>
<td>2</td>
<td>0.177791</td>
<td>0.190741</td>
<td>0.138849</td>
<td>0.224225</td>
<td>0.250333</td>
</tr>
<tr>
<td>3</td>
<td>0.186676</td>
<td>0.103525</td>
<td>0.088887</td>
<td>0.03767</td>
<td>0.098151</td>
</tr>
<tr>
<td>4</td>
<td>0.158891</td>
<td>0.451399</td>
<td>0.178679</td>
<td>0.183349</td>
<td>0.093398</td>
</tr>
<tr>
<td>5</td>
<td>0.233788</td>
<td>0.100544</td>
<td>0.088166</td>
<td>0.173026</td>
<td>0.317712</td>
</tr>
<tr>
<td>6</td>
<td>0.387523</td>
<td>0.395579</td>
<td>0.214217</td>
<td>0.225914</td>
<td>0.316055</td>
</tr>
<tr>
<td>7</td>
<td>0.466542</td>
<td>0.168793</td>
<td>0.161347</td>
<td>0.122675</td>
<td>0.277156</td>
</tr>
<tr>
<td>8</td>
<td>0.650115</td>
<td>0.76848</td>
<td>0.307197</td>
<td>0.148561</td>
<td>0.164106</td>
</tr>
<tr>
<td>9</td>
<td>0.154188</td>
<td>0.127868</td>
<td>0.157942</td>
<td>0.079019</td>
<td>0.115397</td>
</tr>
<tr>
<td>10</td>
<td>0.142897</td>
<td>0.065791</td>
<td>0.04035</td>
<td>0.099944</td>
<td>0.081348</td>
</tr>
<tr>
<td>11</td>
<td>0.030199</td>
<td>0.058407</td>
<td>0.048999</td>
<td>0.04554</td>
<td>0.056951</td>
</tr>
<tr>
<td>12</td>
<td>0.308965</td>
<td>0.084004</td>
<td>0.073236</td>
<td>0.063398</td>
<td>0.115486</td>
</tr>
<tr>
<td>13</td>
<td>0.097312</td>
<td>0.122333</td>
<td>0.08305</td>
<td>0.0679</td>
<td>0.12415</td>
</tr>
<tr>
<td>14</td>
<td>0.180673</td>
<td>0.114616</td>
<td>0.147847</td>
<td>0.108341</td>
<td>0.155632</td>
</tr>
<tr>
<td>15</td>
<td>0.294389</td>
<td>0.162379</td>
<td>0.163469</td>
<td>0.128313</td>
<td>0.204068</td>
</tr>
<tr>
<td>16</td>
<td>0.370703</td>
<td>0.295736</td>
<td>0.39261</td>
<td>0.082624</td>
<td>0.240964</td>
</tr>
<tr>
<td>17</td>
<td>0.089663</td>
<td>0.117827</td>
<td>0.099019</td>
<td>0.113251</td>
<td>0.154837</td>
</tr>
<tr>
<td>18</td>
<td>0.065765</td>
<td>0.07874</td>
<td>0.200776</td>
<td>0.072382</td>
<td>0.078802</td>
</tr>
<tr>
<td>19</td>
<td>0.7826</td>
<td>0.472145</td>
<td>0.181959</td>
<td>0.213217</td>
<td>0.278705</td>
</tr>
<tr>
<td>20</td>
<td>0.138687</td>
<td>0.102879</td>
<td>0.074711</td>
<td>0.112002</td>
<td>0.115839</td>
</tr>
<tr>
<td>21</td>
<td>0.090196</td>
<td>0.138917</td>
<td>0.133495</td>
<td>0.144276</td>
<td>0.111417</td>
</tr>
<tr>
<td>22</td>
<td>0.127351</td>
<td>0.268124</td>
<td>0.130458</td>
<td>0.198688</td>
<td>0.156523</td>
</tr>
<tr>
<td>23</td>
<td>0.337754</td>
<td>0.151276</td>
<td>0.214018</td>
<td>0.113961</td>
<td>0.110098</td>
</tr>
<tr>
<td>24</td>
<td>0.072954</td>
<td>0.079409</td>
<td>0.087274</td>
<td>0.065179</td>
<td>0.089448</td>
</tr>
<tr>
<td>25</td>
<td>0.619115</td>
<td>0.214607</td>
<td>0.317426</td>
<td>0.378104</td>
<td>0.259913</td>
</tr>
</tbody>
</table>
### Table D2  Tunnel Drawing Time

<table>
<thead>
<tr>
<th>Subject</th>
<th>Popup</th>
<th>Popup + Arrows</th>
<th>Forces</th>
<th>Arrows</th>
<th>Popup + Arrows + Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.2846</td>
<td>22.4691</td>
<td>24.5412</td>
<td>19.0123</td>
<td>17.4572</td>
</tr>
<tr>
<td>3</td>
<td>33.3327</td>
<td>16.4571</td>
<td>5.37345</td>
<td>11.5008</td>
<td>5.91523</td>
</tr>
<tr>
<td>4</td>
<td>8.45177</td>
<td>14.46</td>
<td>9.16814</td>
<td>8.79068</td>
<td>8.9318</td>
</tr>
<tr>
<td>5</td>
<td>20.7506</td>
<td>13.1669</td>
<td>17.1672</td>
<td>14.9163</td>
<td>42.9181</td>
</tr>
<tr>
<td>6</td>
<td>58.5433</td>
<td>47.1264</td>
<td>24.2924</td>
<td>22.3755</td>
<td>34.0844</td>
</tr>
<tr>
<td>7</td>
<td>51.4626</td>
<td>46.6347</td>
<td>16.7088</td>
<td>41.6261</td>
<td>56.585</td>
</tr>
<tr>
<td>8</td>
<td>31.4596</td>
<td>41.8355</td>
<td>30.8761</td>
<td>18.334</td>
<td>18.9173</td>
</tr>
<tr>
<td>12</td>
<td>31.5428</td>
<td>37.7938</td>
<td>12.1254</td>
<td>23.1676</td>
<td>24.2925</td>
</tr>
<tr>
<td>13</td>
<td>18.7091</td>
<td>17.7507</td>
<td>11.5837</td>
<td>19.4173</td>
<td>17.8342</td>
</tr>
<tr>
<td>14</td>
<td>19.6842</td>
<td>26.651</td>
<td>34.3251</td>
<td>20.2988</td>
<td>25.1427</td>
</tr>
<tr>
<td>15</td>
<td>31.595</td>
<td>23.3844</td>
<td>30.2799</td>
<td>18.4776</td>
<td>30.3099</td>
</tr>
<tr>
<td>16</td>
<td>56.8643</td>
<td>39.5916</td>
<td>41.0336</td>
<td>26.3264</td>
<td>39.5187</td>
</tr>
<tr>
<td>17</td>
<td>5.94275</td>
<td>6.49207</td>
<td>2.2639</td>
<td>6.03166</td>
<td>4.81083</td>
</tr>
<tr>
<td>18</td>
<td>10.0711</td>
<td>7.62407</td>
<td>8.68948</td>
<td>13.9496</td>
<td>6.52543</td>
</tr>
<tr>
<td>19</td>
<td>30.9125</td>
<td>25.8686</td>
<td>39.2921</td>
<td>24.0042</td>
<td>37.3213</td>
</tr>
<tr>
<td>21</td>
<td>12.3247</td>
<td>13.1175</td>
<td>12.924</td>
<td>14.8154</td>
<td>7.24739</td>
</tr>
<tr>
<td>25</td>
<td>17.4161</td>
<td>7.91816</td>
<td>12.9987</td>
<td>13.2433</td>
<td>6.5404</td>
</tr>
</tbody>
</table>

### Table D3  Encroachment into Danger Zones

<table>
<thead>
<tr>
<th>Subject</th>
<th>Popup</th>
<th>Popup + Arrows</th>
<th>Forces</th>
<th>Arrows</th>
<th>Popup + Arrows + Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.105684</td>
<td>0.078323</td>
<td>0.041863</td>
<td>0.05168</td>
<td>0.039159</td>
</tr>
<tr>
<td>2</td>
<td>0.0922655</td>
<td>0.059334</td>
<td>0.075011</td>
<td>0.249647</td>
<td>0.142678</td>
</tr>
<tr>
<td>3</td>
<td>0.541376</td>
<td>0.264794</td>
<td>0.164911</td>
<td>0.250446</td>
<td>0.148134</td>
</tr>
<tr>
<td>4</td>
<td>0.0787993</td>
<td>0.316616</td>
<td>0.046378</td>
<td>0.181797</td>
<td>0.0475531</td>
</tr>
<tr>
<td>5</td>
<td>0.110587</td>
<td>0.074363</td>
<td>0.080105</td>
<td>0.164612</td>
<td>0.426027</td>
</tr>
<tr>
<td>6</td>
<td>0.338884</td>
<td>0.656944</td>
<td>0.267379</td>
<td>0.286137</td>
<td>0.116197</td>
</tr>
<tr>
<td>7</td>
<td>0.402092</td>
<td>0.125549</td>
<td>0.065946</td>
<td>0.422675</td>
<td>0.0930997</td>
</tr>
<tr>
<td>8</td>
<td>0.370704</td>
<td>1.37528</td>
<td>0.286951</td>
<td>0.211059</td>
<td>0.0839273</td>
</tr>
<tr>
<td>9</td>
<td>0.180459</td>
<td>0.057397</td>
<td>0.074577</td>
<td>0.146269</td>
<td>0.0951836</td>
</tr>
<tr>
<td>10</td>
<td>0.218241</td>
<td>0.098438</td>
<td>0.197812</td>
<td>0.087294</td>
<td>0.14639</td>
</tr>
<tr>
<td>11</td>
<td>0.292957</td>
<td>0.319784</td>
<td>0.19227</td>
<td>0.195968</td>
<td>0.2096</td>
</tr>
<tr>
<td>12</td>
<td>0.325965</td>
<td>0.390427</td>
<td>0.446867</td>
<td>0.232418</td>
<td>0.120604</td>
</tr>
<tr>
<td>13</td>
<td>0.25562</td>
<td>0.144123</td>
<td>0.164869</td>
<td>0.112257</td>
<td>0.394719</td>
</tr>
<tr>
<td>14</td>
<td>0.346306</td>
<td>0.155814</td>
<td>0.137999</td>
<td>0.172371</td>
<td>0.43317</td>
</tr>
<tr>
<td>15</td>
<td>0.609042</td>
<td>0.462227</td>
<td>0.252477</td>
<td>0.377413</td>
<td>0.258754</td>
</tr>
<tr>
<td>16</td>
<td>0.595943</td>
<td>0.246685</td>
<td>1.04495</td>
<td>0.407846</td>
<td>0.698139</td>
</tr>
<tr>
<td>17</td>
<td>0.291147</td>
<td>0.367655</td>
<td>0.110901</td>
<td>0.192685</td>
<td>0.101501</td>
</tr>
<tr>
<td>18</td>
<td>0.390155</td>
<td>0.37053</td>
<td>0.410964</td>
<td>0.246703</td>
<td>0.198908</td>
</tr>
<tr>
<td>19</td>
<td>0.704901</td>
<td>0.138154</td>
<td>0.169663</td>
<td>0.527652</td>
<td>0.171262</td>
</tr>
<tr>
<td>20</td>
<td>0.464967</td>
<td>0.239087</td>
<td>0.215188</td>
<td>0.390187</td>
<td>0.414079</td>
</tr>
<tr>
<td>21</td>
<td>0.339453</td>
<td>0.355444</td>
<td>0.184547</td>
<td>0.215819</td>
<td>0.0945235</td>
</tr>
<tr>
<td>22</td>
<td>0.422267</td>
<td>0.180412</td>
<td>0.249953</td>
<td>0.345202</td>
<td>0.0963684</td>
</tr>
<tr>
<td>23</td>
<td>0.218786</td>
<td>0.400889</td>
<td>0.120052</td>
<td>0.199973</td>
<td>0.060949</td>
</tr>
<tr>
<td>24</td>
<td>0.3797787</td>
<td>0.268306</td>
<td>0.161611</td>
<td>0.167814</td>
<td>0.213799</td>
</tr>
<tr>
<td>25</td>
<td>0.309215</td>
<td>0.021514</td>
<td>0.066613</td>
<td>0.181008</td>
<td>0.281186</td>
</tr>
</tbody>
</table>
Appendix E : Residuals

Figure E1 Residuals of extra distance drawn. In the fitted-value plot the spread of residuals is uneven for untransformed, but becomes reasonably uniform for transformed.
Figure E2 Residuals of time. In the fitted-value plot the spread of residuals is uneven for untransformed, but becomes reasonably uniform for transformed.
Figure E3 Residuals of danger violation. In the fitted-value plot the spread of residuals is uneven for untransformed, but becomes reasonably uniform for transformed.
Appendix F : 
Analysis of Variation Results

### Table F1 Analysis of variance for excess tunnel distance

<table>
<thead>
<tr>
<th>Source of variation</th>
<th>d.f.</th>
<th>s.s.</th>
<th>m.s.</th>
<th>v.r.</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject (not adjusted for Order.Treatment)</td>
<td>24</td>
<td>33.205</td>
<td>1.3837</td>
<td>8.62</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Order</td>
<td>4</td>
<td>0.2916</td>
<td>0.0729</td>
<td>0.45</td>
<td>0.786</td>
</tr>
<tr>
<td>Treatment</td>
<td>4</td>
<td>1.8146</td>
<td>0.4536</td>
<td>2.83</td>
<td>0.030</td>
</tr>
<tr>
<td>Order.Treatment (adjusted for Subject)</td>
<td>16</td>
<td>1.1724</td>
<td>0.0733</td>
<td>0.46</td>
<td>0.960</td>
</tr>
<tr>
<td>Residual</td>
<td>80</td>
<td>12.8426</td>
<td>0.1605</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table F2 Analysis of variance for tunnel drawing time

<table>
<thead>
<tr>
<th>Source of variation</th>
<th>d.f.</th>
<th>s.s.</th>
<th>m.s.</th>
<th>v.r.</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject (not adjusted for Order.Treatment)</td>
<td>24</td>
<td>34.0756</td>
<td>1.4198</td>
<td>27.41</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Order</td>
<td>4</td>
<td>1.4779</td>
<td>0.3695</td>
<td>7.13</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Treatment</td>
<td>4</td>
<td>1.3308</td>
<td>0.3327</td>
<td>6.42</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Order.Treatment (adjusted for Subject)</td>
<td>16</td>
<td>0.9029</td>
<td>0.0564</td>
<td>1.09</td>
<td>0.379</td>
</tr>
<tr>
<td>Residual</td>
<td>80</td>
<td>4.1400</td>
<td>0.0518</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table F3 Analysis of variance for danger encroachment

<table>
<thead>
<tr>
<th>Source of variation</th>
<th>d.f.</th>
<th>s.s.</th>
<th>m.s.</th>
<th>v.r.</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject(not adjusted for Order.Treatment)</td>
<td>24</td>
<td>21.5037</td>
<td>0.8960</td>
<td>3.08</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Order</td>
<td>4</td>
<td>0.1803</td>
<td>0.0451</td>
<td>0.15</td>
<td>0.960</td>
</tr>
<tr>
<td>Treatment</td>
<td>4</td>
<td>5.6808</td>
<td>1.4202</td>
<td>4.88</td>
<td>0.001</td>
</tr>
<tr>
<td>Order.Treatment (adjusted for Subject)</td>
<td>16</td>
<td>2.1512</td>
<td>0.4249</td>
<td>1.46</td>
<td>0.137</td>
</tr>
<tr>
<td>Residual</td>
<td>80</td>
<td>23.2990</td>
<td>0.2912</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix G:

Ability-to-Ignore-Feedback Results

Table H1 Ability-to-ignore-advice results

<table>
<thead>
<tr>
<th>Subject</th>
<th>Popup</th>
<th>Popup + Arrows</th>
<th>Forces</th>
<th>Arrows</th>
<th>Popup + Arrows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.016321026</td>
<td>0.151284147</td>
<td>0.027405</td>
<td>0.014373</td>
<td>0.001476</td>
</tr>
<tr>
<td>2</td>
<td>0.062887435</td>
<td>0.239575823</td>
<td>0.002878</td>
<td>0.25908</td>
<td>0.20094</td>
</tr>
<tr>
<td>3</td>
<td>0.6991556</td>
<td>0.573201983</td>
<td>0.193132</td>
<td>0.117173</td>
<td>0.122299</td>
</tr>
<tr>
<td>4</td>
<td>0.034642357</td>
<td>0.25233807</td>
<td>0.063204</td>
<td>0.263777</td>
<td>0.689036</td>
</tr>
<tr>
<td>5</td>
<td>0.370698018</td>
<td>0.366635</td>
<td>0.659755</td>
<td>0.435422</td>
<td>0.156767</td>
</tr>
<tr>
<td>6</td>
<td>0.770363029</td>
<td>0.595035642</td>
<td>0.271925</td>
<td>0.537459</td>
<td>0.947405</td>
</tr>
<tr>
<td>7</td>
<td>0.639584454</td>
<td>0.480749168</td>
<td>0.45422</td>
<td>0.263777</td>
<td>0.182558</td>
</tr>
<tr>
<td>8</td>
<td>0.390584922</td>
<td>0.131377</td>
<td>0.08093</td>
<td>0.467529</td>
<td>0.037736</td>
</tr>
<tr>
<td>9</td>
<td>0.373984906</td>
<td>0.331077017</td>
<td>0.218689</td>
<td>0.368263</td>
<td>0.223529</td>
</tr>
<tr>
<td>10</td>
<td>0.490801452</td>
<td>0.443541536</td>
<td>0.850214</td>
<td>0.850849</td>
<td>0.263777</td>
</tr>
<tr>
<td>11</td>
<td>0.469180971</td>
<td>0.803450945</td>
<td>0.492217</td>
<td>0.689036</td>
<td>0.037736</td>
</tr>
<tr>
<td>12</td>
<td>0.46625122</td>
<td>0.168425438</td>
<td>0.216221</td>
<td>0.437112</td>
<td>0.156767</td>
</tr>
<tr>
<td>13</td>
<td>0.74044752</td>
<td>0.549472</td>
<td>0.812558</td>
<td>0.467529</td>
<td>0.037736</td>
</tr>
<tr>
<td>14</td>
<td>0.526087961</td>
<td>0.481879386</td>
<td>0.423435</td>
<td>0.473057</td>
<td>0.648019</td>
</tr>
<tr>
<td>15</td>
<td>0.293454716</td>
<td>0.09380551</td>
<td>0.081969</td>
<td>0.055694</td>
<td>0.183023</td>
</tr>
<tr>
<td>16</td>
<td>0.079824054</td>
<td>0.069595055</td>
<td>0.007287</td>
<td>0.225365</td>
<td>0.008184</td>
</tr>
<tr>
<td>17</td>
<td>0.10857211</td>
<td>0.030791883</td>
<td>0.047218</td>
<td>0.056479</td>
<td>0.037736</td>
</tr>
<tr>
<td>18</td>
<td>0.003167271</td>
<td>0.007535549</td>
<td>0.020447</td>
<td>0.003443</td>
<td>0.431887</td>
</tr>
<tr>
<td>19</td>
<td>0.048740601</td>
<td>0.04873504</td>
<td>0.028288</td>
<td>0.04205</td>
<td>0.053896</td>
</tr>
<tr>
<td>20</td>
<td>0.066998216</td>
<td>0.029404977</td>
<td>0.040978</td>
<td>0.057081</td>
<td>0.032263</td>
</tr>
<tr>
<td>21</td>
<td>0.024360871</td>
<td>0.027761632</td>
<td>0.039331</td>
<td>0.045166</td>
<td>0.036244</td>
</tr>
<tr>
<td>22</td>
<td>0.041704865</td>
<td>0.035659811</td>
<td>0.030522</td>
<td>0.044706</td>
<td>0.038932</td>
</tr>
<tr>
<td>23</td>
<td>0.015781164</td>
<td>0.015952649</td>
<td>0.03528</td>
<td>0.015454</td>
<td>0.051903</td>
</tr>
<tr>
<td>24</td>
<td>0.186842169</td>
<td>0.686556069</td>
<td>0.018172</td>
<td>0.244197</td>
<td>0.051903</td>
</tr>
</tbody>
</table>

Analysis of Variation for ability to ignore advice

<table>
<thead>
<tr>
<th>Source of variation</th>
<th>d.f.</th>
<th>s.s.</th>
<th>m.s.</th>
<th>v.r.</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>24</td>
<td>312.302</td>
<td>13.013</td>
<td>12.88</td>
<td>&lt;.001</td>
</tr>
<tr>
<td>Order</td>
<td>4</td>
<td>19.677</td>
<td>4.919</td>
<td>4.87</td>
<td>0.002</td>
</tr>
<tr>
<td>Treatment</td>
<td>4</td>
<td>13.578</td>
<td>3.394</td>
<td>3.36</td>
<td>0.014</td>
</tr>
<tr>
<td>Order.Treatment (adjusted for Subject)</td>
<td>16</td>
<td>24.469</td>
<td>1.529</td>
<td>1.51</td>
<td>0.120</td>
</tr>
<tr>
<td>Residual</td>
<td>64</td>
<td>64.612</td>
<td>1.010</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix H :

Acronyms

CSIRO Commonwellth Scientific Industrial Research Organisation (Australia)
NLM National Library of Medicine (USA)
SIGGRAPH Special Interest Group, Graphics
MMVR Medicine Meets Virtual Reality
DOF Degrees of Freedom (sometimes also used for Degrees of Force)
IEEE Institute of Electrical and Electronics Engineers
TCP Transmission Control Protocol
UDP User Datagram Protocol
HCI Human Computer Interaction
API Application Programming Interface