A clinically valid simulator with tactile sensing to train specialists to perform cochlear implantation

Catherine A. Todd
University of Wollongong

A Clinically Valid Simulator with Tactile Sensing to Train Specialists to perform Cochlear Implantation

by

Catherine Angela Todd

B. Eng. Electrical (Hons.)

This thesis is submitted in fulfilment of the requirements for award of the PhD degree from the University of Wollongong, Australia

School of Electrical, Computer and Telecommunications Engineering

Faculty of Informatics

June, 2006

"*I saw the angel in the marble*

and carved until I set him free."

~ Michelangelo (1475 - 1564)

DECLARATION

I hereby declare that I, Catherine Angela Todd, am the sole author of this dissertation. I declare that the material contained within this thesis is my own work and that referenced work is appropriately cited. I also declare that I am not aware of any similar work undertaken prior to the submission of this thesis.

Catherine Angela Todd

June 2006

ACKNOWLEDGEMENTS

This PhD thesis could not have been pursued without the undeviating support from family, friends and work colleagues, which has been ongoing for over three years. The facilities and funding provided by the University of Wollongong and Industrial Partner, Cochlear™, have enabled me to achieve project outcomes using optimal resources.

First and foremost I would like to thank my thesis supervisor, Associate Professor Fazel Naghdy for his unending support, guidance and commitment. Fazel has a remarkably high level of professionalism to which I aspire and I am greatly appreciative of the academic and moral support he has provided me throughout the course of my PhD. The success of completing the thesis I owe to my supervisor.

I would like to thank my parents, Liane and Ray, for the emotional support they provided, particularly during times of asperity. Similarly, I thank my many friends for their supportive words, listening ears and positive encouragement. I also appreciate the support offered by my work colleagues, including fellow thesis students, academics, administrative and general staff in the School of Electrical, Computer and Telecommunications Engineering at the University of Wollongong. I give a special thanks to the administration ladies, Ros, Tracey and Maree, and the workshop staff for their excellent technical support.

Academic work is unable to be complete without financial aid. This work was supported in part by the Australian Research Council (ARC) under a Linkage Grant, as an Australian Postgraduate Award (Industry) offered by the University of Wollongong. I

would like to thank the Industrial Partner, Cochlear™ and staff. It has been a pleasure working at the University of Wollongong under impeccable guidance and resources.

I am truly grateful to be given the opportunity to become involved in this interesting area of PhD research and am ever thankful to those who have helped me achieve it.

Catherine Angela Todd

June 2006

ABSTRACT

Cochlear implantation is a maximally invasive surgical procedure aimed at overcoming human inaudibility and providing the sensation of sound to thousands of severely deaf recipients worldwide. Specialists require extensive training to perform the surgery, yet traditional approaches such as cadaver dissection and device insertion can prove costly. Alternative training schemes have not been developed, however surgical simulators that offer force feedback during anatomical model manipulation may provide the answer.

In the work, a novel approach to medical education is presented. It combines haptic technology and computer visualisation to recreate cochlear implantation in a virtual environment. The surgical simulator provides visual and haptic rendering during cochlear implant insertion into a virtual model of the human Scala Tympani. As the user inserts the sub-sampled array into a three-dimensional, reproducible representation of the Scala Tympani, collisions between the electrode and Scala Tympani walls are detected. In response, real-time forces are delivered back through the haptic device in a closed loop control system.

Insertion studies are performed to evaluate the cochlear implant insertion process. Electrode array trajectories and output forces are monitored during device insertion into a synthetic model of the Scala Tympani. The force, torque and position data produced from the experiments are used in the final stage of work for simulator validation.

A three-dimensional, surface description of the human Scala Tympani is derived from measured data and parameterised for future reproduction. It is visualised in a virtual environment, the Reachin Application Programming Interface, where visual and haptic rendering is implemented to make the insertion process interactive. Algorithms are

produced and program optimisations performed to enable real-time, dynamic manipulation of the environment. Real world physical attributes are added to the Scala Tympani surfaces and electrode carrier to make the scene more realistic.

System validation is performed by statistical and qualitative comparisons between the force profiles produced from the simulation and experimentation. The results are presented and evaluated in terms of overall system performance.

The thesis offers unique approaches for simulator design, development and validation. The significant contributions of the work are reported, as are the benefits, with recommendations for future system enhancements.

THESIS PUBLICATIONS

1. C. Todd and F. Naghdy, “Virtual Cochlear Implantation”, to be presented at the 9th International Conference on Cochlear Implants, Hofburg, Vienna, Austria, 2006.

2. C. Todd and F. Naghdy, “Cochlear Implant Insertion: A Virtual Approach for Medical Education”, presented at the 6th Annual International Meeting on Medical Simulation (IMMS 2006), pp 564-567, San Diego, California, United States of America, Society for Medical Simulation, 2006.

3. C. Todd and F. Naghdy, “Virtual Cochlear Implant Insertion for Medical Education”, presented at World Haptics Conference (WHC 2005), Piza, Italy, IEEE Computer Society, 2005.

4. C. Todd and F. Naghdy, “Parametric Model of the Scala Tympani for Haptic-Rendered Cochlear Implantation”, presented at Medicine Meets Virtual Reality 13 (MMVR 2005), pp 564-567, Long Beach, California, United States of America, IOS Press, Netherlands, 2005.

5. C. Todd and F. Naghdy, “Visual and Haptic Rendering of the Temporal Bone for Cochlear Implant Surgery”, presented at the 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004), Orlando, Florida, United States of America, International Institute of Informatics and Systemics, 2004.

6. C. Todd, F. Naghdy and Stephen O'Leary, "Geometric Modeling of the Temporal Bone for Cochlear Implant Simulation", presented at SPIE Medical Imaging (SPIE 2004), vol. 5367, pp 482-490, San Diego, California, United States of America, Proceedings of SPIE (SPIE, Bellingham, WA), 2004.

7. C. Todd and F. Naghdy, "Visual and Haptic Rendering of Temporal Bone Surgery", presented at 3rd IFAC Symposium on Mechatronic Systems (IFAC 2004), pp 487-492, Manly, Australia, IFAC (Computer Proceedings), 2004.

8. C. Todd, F. Naghdy and S. O'Leary, "Geometric Modelling of a Haptic Rendered Computer Simulator for Cochlear Implantation", presented at the Second International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2004), pp PS01-3-03, Singapore, National University of Singapore (Computer Proceedings), 2004.

9. C. Todd and F. Naghdy, "Force Application during Cochlear Implant Insertion: An Analysis for Improvement of Surgeon Technique", IEEE Engineering in Medicine and Biology Society, (submitted to, February 2006).

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

F_F	Force Due to Friction
F_{in}	Input Force
F_n	Normal Force
μ	Coefficient of Friction
3D	Three Dimensional
API	Application Programming Interface
AOS	Advance Off- Stylet
BM	Basilar Membrane
CI	Cochlear Implant
CP	Cochlear Partition
CT	Computed Tomography
DOF	Degree of Freedom
ENT	Ear, Nose and Throat
FEM	Finite Element Model
GUI	Graphical User Interface
IW	Inner Wall
MFS	Method of Finite Spheres
MIP	Maximum Intensity Projection
MRI	Magnetic Resonance Imaging
MRM	Magnetic Resonance Microscopy
OC	Organ of Corti
OSL	Osseous Spiral Lamina

OW	Outer Wall
ROI	Region of Interest
RW	Round Window
SFM	Scanning Force Microscopy
SIT	Standard Insertion Technique
SM	Scala Media
ST	Scala Tympani
SV	Scala Vestibuli
VBA	Visual Basic for Applications
VE	Virtual Endoscopy
VRML	Virtual Reality Modelling Language
VTK	Visualization Toolkit

Chapter 1

INTRODUCTION

1.1 Overview

Virtual surgical suites are becoming a reality. Rapid advances in computer technology, including the development of dedicated software and specialised hardware systems, have facilitated the evolution of surgical simulators that may add a new dimension to medical education. More recent designs provide the user with touch, sight and sound sensation associated with complicated medical procedures. Such simulations can supplement traditional approaches to surgeon training. Three-dimensional reconstructions of the human body are becoming more realistic, both in visual representation and physical behaviour. It is now possible to replicate complex structures with real-world attributes and interact with them in real time. With the advent of haptic interfaces and the refinement of haptic algorithms, the prospect of including surgical simulators in medical school programs for clinical training and assessment is apparent. Applications in this field are numerous, as are the benefits, and these will be discussed in detail in the following sections of work.

Despite the recent success that has been enjoyed in the field of surgical simulation, it is still a new field of research and there remain areas that warrant attention and development. Of particular interest are maximally invasive procedures that require extensive surgeon training; entailing years of study, practise and financial cost. Clinical interventions of this type can put the patient at risk of trauma or injury. It is therefore of

benefit to both the specialist and patient if the procedure can be recreated in a virtual environment for pre-operative evaluation and surgeon practice. Virtual environments are becoming viable, low-cost options for training specialists who undertake high risk medical procedures. The field of surgical simulation is vast and there are a variety of approaches to the visual and haptic rendering of anatomical structures within these virtual training suites. System constraints often limit the capabilities of the design. The surgical procedure itself will vary significantly between operations. The technique that is implemented must be evaluated for desired system functionality and verified using quantitative methods. A surgical simulator should not only mimic highly specialised surgical practices, but be applicable to other areas of study.

In this work, a clinically valid surgical simulator with haptic feedback has been developed to train specialists in cochlear implant (CI) insertion. This is a highly invasive medical procedure that requires a considerable level of surgical skill. Despite the specialised field of expertise, the approaches taken in this research can be applied to other areas of surgical simulation. The thesis provides detail of the overall process, including the stages of simulator design, implementation and validation. The next section of work describes the conditions and procedure associated with cochlear implantation, including fundamental cochlear mechanics, implant design and functionality, the surgical procedure and current tuition schemes that are available to the training otologist. The concept of surgical simulation, including the stages of realisation and the benefits of such systems, is introduced. Objectives of the research are clearly identified to show the significant contribution of the work and the scope of the thesis is presented.

1.2 The Human Cochlea

The human cochlea is a three-dimensional spiral with 2 ½ [1, 2] to 2 ¾ [3-6] turns. It has a maximum diameter of 5mm to 6mm and is located within the petrous quadrant of the temporal bone, deep within the inner ear. It is anatomically complex and varies between individuals, in both shape (such as intra-cochlear tilt [7, 8]) and size (including cochlear length [3, 8-11] and diameter [11]). Its primary function is to transduce mechanical motion into electrical signals that trigger neural activity, which the brain perceives as sound. The small, delicate structure contains two canals filled with incompressible fluid: the Scala Tympani (ST) and Scala Vestibuli (SV), which meet at the most apical part (helicotrema) of the cochlea [1]. Sandwiched between these two tapering chambers is the Cochlear Partition (CP): a complex array of tissues housing the Scala Media (SM), which contains endolymph fluid [1, 12]. The Basilar Membrane (BM) comprises the inferior boundary of the SM, with Reissner's Membrane forming the roof of the triangular structure [1]. The BM is held to the outer and inner walls of the cochlea by the spiral ligament and Osseous Spiral Lamina (OSL), respectively [1]. The entire structure is encased in a bone termed the otic capsule [12]. A cross-section of the human cochlea is shown in Figure 1.1 and the cochlear spiral is shown in Figure 1.2 with its position relative to the rest of the human ear.

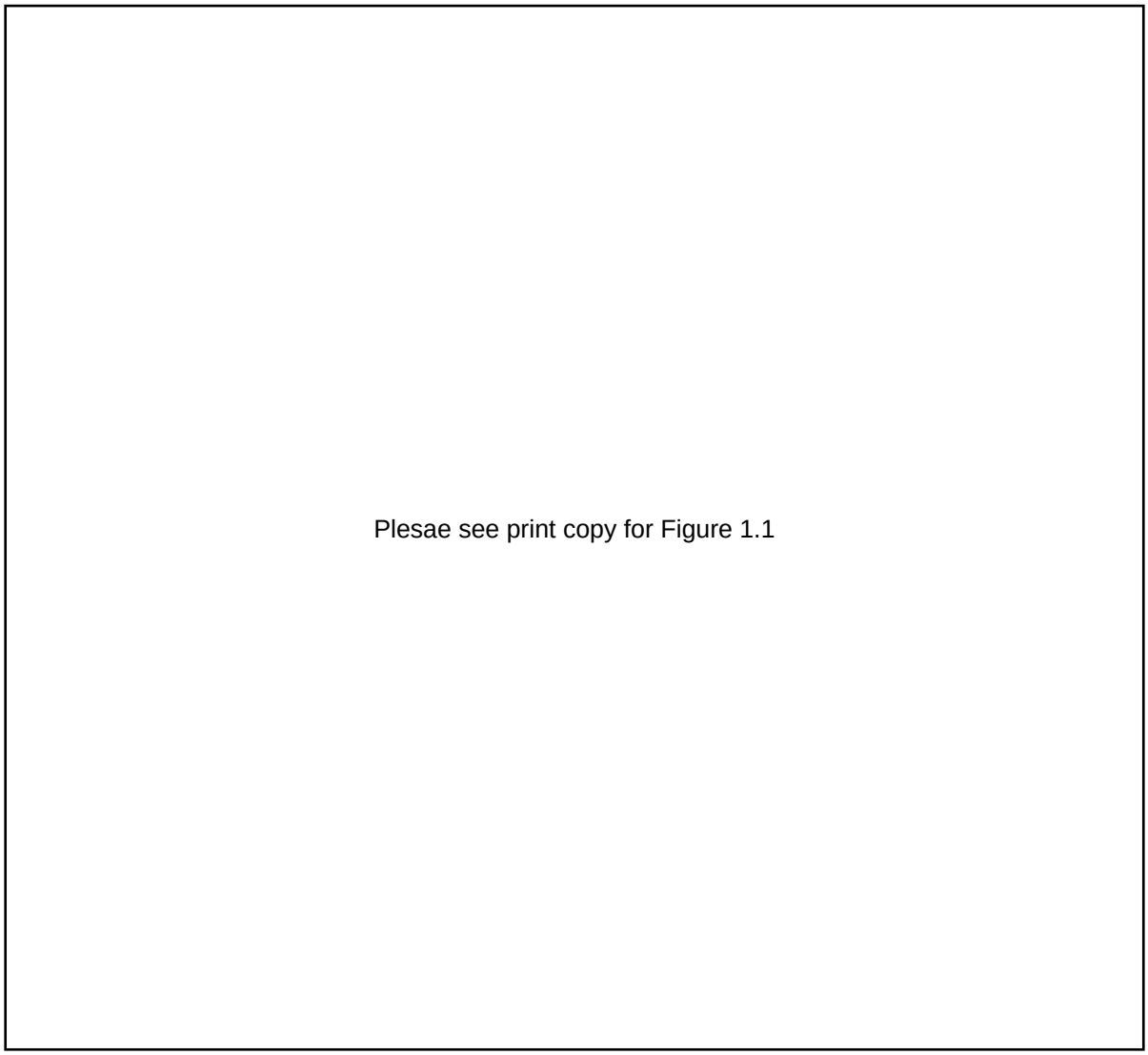

Figure 1.1. A cross-section of the cochlea, captured in a transverse plane [2]. The image shows the three fluid-filled chambers within the otic capsule: the inferior Scala Tympani (ST), the superior Scala Vestibuli (SV) and the medial Scala Media (SM) contained within the membranous Cochlear Partition (CP). The figure shows the inner hair cells which rest against the Basilar Membrane (BM) and bend upon fluid vibrations to cause electrochemical stimulation of the auditory nerve.

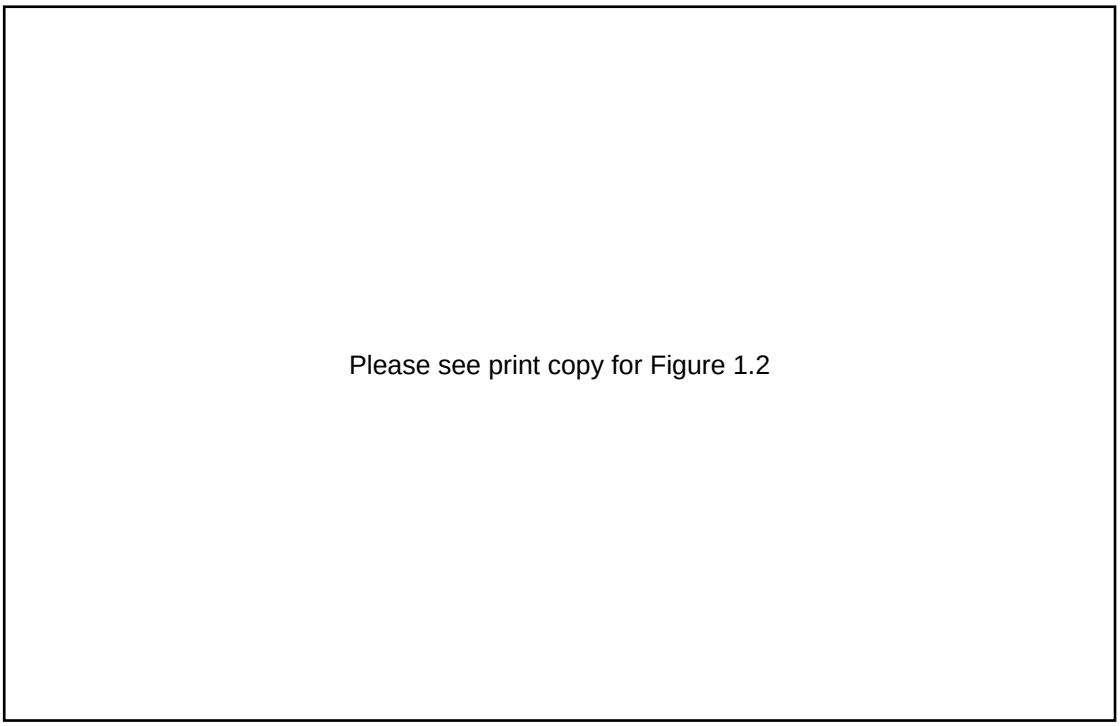

Figure 1.2. The cochlear spiral, located deep within the inner ear (as shown in the figure) [13]. It is located in the petrous quadrant of the temporal bone and access to the Round Window (RW) is gained via a mastoidectomy.

Extending from its base to apex along the cochlear periphery are clusters of tiny hair cells (Figure 1.1). When air pressure waves captured by the external ear trigger mechanical vibrations in the fluid of the cochlea, these hair cells oscillate at dedicated frequency bands to stimulate the auditory nerve. The inner hair cells are primarily responsible for audition and are located along the length of the BM, about the modiolar (central spiral) axis. Hair cell stereocilia movement at the base results in high frequency perception whilst apical regions correspond to low frequency sounds [12]. Degeneration of hair cells is the most common cause of deafness [14]. This can result from over-exposure to loud noise, typically over 90dB for long-term exposure or 140dB for short blasts [15]. Clinical conditions can also destroy these cells and can be prenatal (such as congenital rubella and syphilis) or acquired (for example, bacterial meningitis [16] and

Meniere's disease [14]). The process of aging is also shown to cause hair cell deterioration [1, 2, 17]. Damage to this interface may lead to profound hearing impairment as the auditory nerve can no longer be excited within these region/s [14]. Sensorineural deafness affects a high percentage of the global population. In Australia it is estimated that 17% of the community suffer some degree of hearing loss [18].

1.3 Cochlear Implantation

Cochlear implants are prosthetic medical devices that are designed to overcome the electro-neural deficiency caused by inner hair cell deterioration and facilitate the perception of sound by replacing mechanical stimulus of the auditory nerve with electrical excitation. It is widely accepted that cochlear implants are the only device that can restore hearing to severe or profoundly deaf recipients who gain no benefit from a hearing aid [16]. The hardware consists of a microphone to detect the sound, a speech processor which interprets and converts the sound into electrical signals, a signal transmitter and an internal implant [14]. The implantable part of the device contains an electrode array embedded in a silicone carrier (Figure 1.3). It is designed to be inserted into the human ST [14, 16, 19-22] (Figure 1.4), with the SV as a secondary passage for implantation [16, 20, 23]. Multi-channel electrode arrays are designed to assume a final position along the cochlea and the electrodes are exposed for individual stimulation of particular subsets of the audible frequency range. For humans, this is 20Hz to 20 kHz. Since the first successful cochlear implantation in 1978 by Professor Graeme Clark, over 60,000 people worldwide have been fitted with the device. Recipients range in age, from as young as 12 months [24, 25] to 94 years and degree of success is also variable [14, 16].

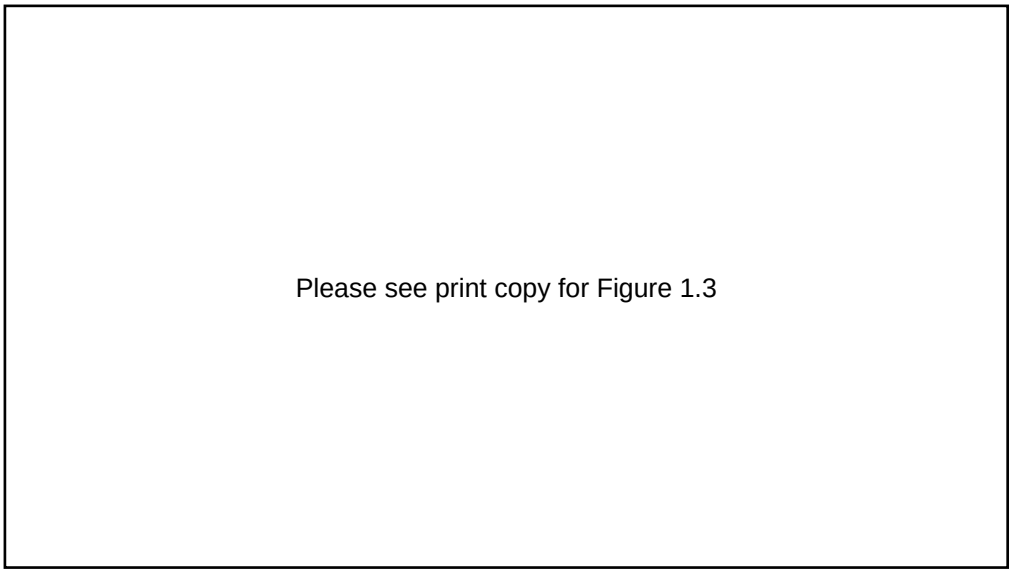

Figure 1.3. Nucleus® 24 Contour™ electrode (picture provided by Cochlear™ 2005). The Contour is shown in its pre-curled state, after partial stylet withdrawal.

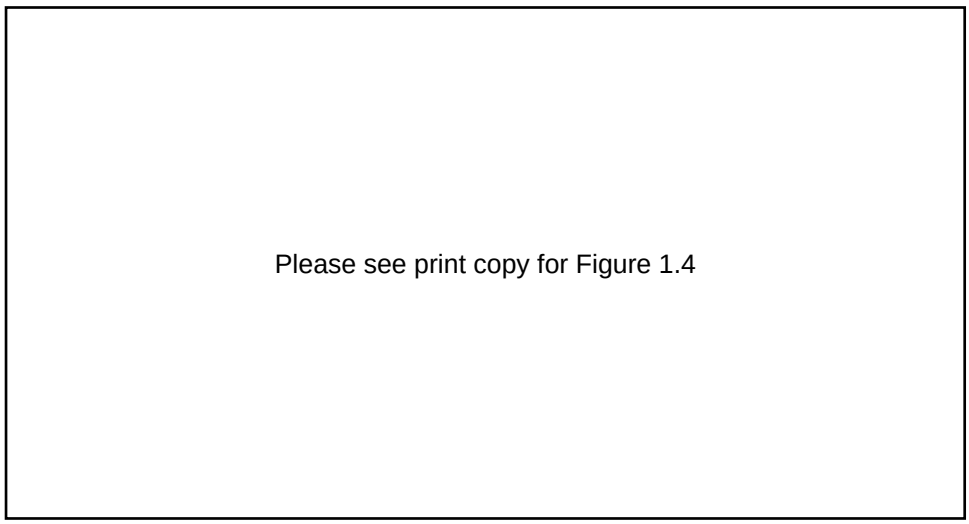

Figure 1.4. The three-dimensional cochlear spiral, with an electrode array inserted into the Scala Tympani (ST) chamber [26].

The complete process associated with cochlear implantation is a particularly intrusive one, as the surgeon progresses from the external to inner ear through a myriad

of delicate anatomical structures which can be affected by clinical anomalies (such as dysplasia or ossification of the cochlea [16]). Care must be taken to avoid inducing extensive trauma or loss of motor-sensory function within this region. Patient variability makes this task more difficult. Although human cochleae exhibit similarities in shape, there are individual differences in cochlear geometry (including length, intra-cochlear tilt and cochlear canal radii [8]).

Prior to prosthetic insertion, the surgeon typically acquires access to the Round Window (RW) of the cochlea in a three-stage, stepwise process, which includes a mastoidectomy, cochleostomy and the cochlear device implantation. Following initial anaesthetisation of the patient, an incision is made into the skin above and behind the external ear [16], to reveal muscle which is also incised. A mastoidectomy is then performed. This involves dissection of the mastoid quadrant of the temporal bone via drilling with irrigation (as a coolant) and suction of the bone dust. Once the RW is revealed, a cochleostomy is performed at this site to provide an entrance for insertion of the electrode. Forceps are used by the Ear, Nose and Throat (ENT) surgeon to grasp the end of the electrode carrier and for its advancement into the ST (as the primary passage for insertion) [16, 23, 25]. When the electrode is in place, a straightening stylet (if in existence) is removed by its end loop and connective tissue is placed around the array at the cochleostomy site to seal it [16] and prevent post-operative device slippage [27].

Following surgery, audiologists facilitate the development of a recipient's speech recognition; examining communication, vocal, speech and language skills [14]. A person's performance gives an indication as to their level of hearing. Collectively, these results signify considerable variability between individuals in their ability to perceive and interpret sound, following the implantation. Some patients can decipher audio signals without the need for lip-reading whilst others require this assistance [14,

16, 28]. Success rates for people receiving cochlear implants will vary depending on cochlear condition and medical history of the patient. Post-lingually deaf persons generally achieve higher levels of speech intelligibility than the pre-lingually deaf [14, 16, 29]. The period over which the post-lingually deaf have suffered hearing loss and the time-frame of hearing also influence results [14, 16]. Usually, people who have had short-term hearing loss after being able to hear for a lengthened period achieve greater success than those who experience long-term hearing loss with less interpretation of sound [14, 29]. Age of onset of deafness as well as age of implantation and duration of usage have also been shown to affect the outcome [14].

Cochlear implants are shown to be cost-effective and improve a person's quality of life, providing benefit to the recipient and community as a whole [16]. Contraindications such as the Michel deformity (no cochlea) may prevent implantation [16]. Degree of deafness will be another consideration for determining the suitability of a candidate for implantation. The number of surviving spiral ganglion cells directly restricts a recipient's ability to perceive sound [14]. Whilst these influences cannot be altered, controllable factors such as electrode placement, orientation and technique for insertion of the device may be enhanced.

Poor technique for implantation may result in ineffective operation of the device or induce trauma to the delicate cochlea structure itself [16]. Insertion studies have shown that the electrode array should be positioned close to the modiolus and oriented towards it to achieve optimal sound perception [14, 25, 30-34]. Insertion depth will also affect perceptible frequency bandwidth [9, 25, 35] and all bands from base (high frequencies) to apex (low frequencies) should be audible. Force administration will also have an affect on final electrode position [36-38], likelihood of trauma in the region of the Basal turn [35-37, 39] and damage to the electrode itself [16, 37, 39]. Carrier

stiffness [40, 41], restoration forces [31] and inter-electrode spacing [8, 14] can also affect the degree of success for the CI recipient. For correct electrode placement and surgical technique, the surgeon is expected to undertake extensive training in this specialisation.

1.4 Insertion Studies

Insertion studies reveal the importance of appropriate surgical technique for correct electrode placement and the minimisation of force delivery during a cochlear implantation. Research in this area has shown that excessive force delivery during electrode advancement should be avoided [23] as it can directly induce extensive intracochlear trauma [11, 25, 30, 31, 35-39, 42-44] and lead to undesirable placement of the array [36-38]. Attempts to minimise these forces and ideally position the array have involved the development of electrode carrier positioners [6, 30, 37, 45], modifications to surgeon technique such as change in size [38] or location [38, 46] of the cochleostomy and alterations in design, mechanical properties, trajectory and orientation of the implant (for example, development of the perimodiolar electrode).

1.4.1 Insertion Trauma

During the insertion, the otologist should aim to avoid damage to both the membranous layers that reside within the cochlear spiral and the delicate electrode array. Poor implantation affected by angle of approach, electrode orientation and/or excessive force administration may cause damage to the membranous CP, such as distortion or penetration of the spiral ligament [11, 30, 36, 37, 39, 40, 43, 44, 47], OSL [35-37, 39, 40, 43, 47], tearing of Reissner's Membrane [37] or piercing of the BM [30, 31, 35-40, 43, 44]. Insertion trauma of this type can lead to loss of spiral ganglion cells and hence

neural degeneration [31, 35, 37, 39]. Insertion studies have shown that surgeon technique can cause this: if the electrode is advanced beyond the point of first resistance, which is in the area of the Basal turn, it is likely that this type of damage will result [33, 37, 39]. Change in orientation of the electrode or partial withdrawal of the stylet (for the Contour array) at this point is recommended [31, 38, 43, 44]. Recognition of this point of contact is conveyed to the otologist purely through the sense of touch. Delivering force feedback to the user throughout this procedure is therefore essential in order to prevent trauma and damage to the array, as well as for optimal electrode positioning.

There is also concern that large restoration forces in perimodiolar designs might instigate damage along the modiolar wall [31]. Tearing of the CP may result in the array assuming a final position inside the SV as opposed to a desired location along the ST inner wall [31, 36-38, 40, 43, 47]. For a worst case scenario, insertion trauma may render the device ineffective, particularly if there is significant loss of the spiral ganglion cells since there will be nothing for the implant to stimulate.

1.4.2 Implant Position

Correct placement of the implant is imperative to achieve optimal device performance [23]. Ideally, the electrode array should lie within close proximity and be oriented toward the modiolus in order to stimulate the auditory neurons in this region [14, 25, 30-34, 40, 48]. Perimodiolar positioning of the electrodes can decrease stimulating current levels [6, 49], reduce channel interference and stimulate discrete clusters of spiral ganglion cells [25, 31, 32, 34, 35]. A deep insertion is desirable [3, 14, 16, 21, 25, 30, 31, 35] and allows for stimulation of neurons at the base to more apical locations, thus covering the entire cochlear frequency range [21]. Consequently, frequency

coverage and overall sound quality perceived by the recipient is affected by the final position of the electrode array [3, 14, 25]. Contour and Contour Advance electrodes have marker ribs to provide the surgeon with an indication of reasonable insertion depth (Figure 1.3). The latter also has a white line on the silicone housing, located approximately 9mm from the tip of the implant, to aid in the Advance Off- Stylet (AOS) insertion.

1.4.3 Damage to the Implant

The implant itself is extremely small and highly flexible. Excessive force delivery can cause deformation of the electrode [16] which may affect its functionality and possibly its trajectory. Contour and Contour Advance arrays are approximately 22mm in length (from the tip to the first marker rib), tapering from a tip diameter of about 0.5mm (0.4mm Advance) to 0.8mm. The silicone-coated electrode is kept straight by a thin platinum wire (stylet). Stylet withdrawal will result in the array returning to its pre-curled state. Array and/or stylet deformation is possible during an insertion and may involve the electrode carrier buckling (primarily at the tip) upon collision with the ST opening or walls [37, 39, 42], in the vicinity of the Basal turn. Array deformation may require re-insertion of a new electrode [31] which is costly, or may lead to undesirable outcomes following surgery. Damage to the electrodes themselves may result in channel malfunction and inaudibility of certain frequency bands. After significant deformation, the carrier has reduced capacity to return back into its geometry. This can have a negative influence on final electrode placement.

Research findings show that correct electrode placement, the minimisation of insertion trauma and the careful handling of the implant are essential for optimal device performance. Despite the results that indicate an apparent relationship between surgical

technique (including force delivery) and outcome for the patient (such as trauma or electrode position), there has been a lack of investigation into force output during a CI insertion. In Chapter 3 this will be further investigated. The study is intended to contribute to this limited area of research as well as to provide insertion force information that will be used to validate the surgical simulator produced in this work. The simulator is expected to provide the surgeon with a virtual environment in which he or she can practice this delicate, maximally invasive procedure. Surgical simulators of this type are useful in familiarising the surgeon with the correct force application during a prosthetic implantation that will ideally position the implant with minimal risk of trauma or injury to the anatomical structures that reside within the region.

1.5 Surgical Training Schemes

ENT specialists undergo extensive training to perform CI surgery. In vivo implantation is only performed by skilled otologists. However, surgeons progressively work towards this stage under the guidance of a mentor. Practical experience is gained in vitro by cadaver exploration, dissection and cochlear implantation.

Despite the surgical complexity of the implantation procedure, educational resources for medical instruction in cochlear implantation are limited to temporal bone drilling laboratories, observation of experienced ENT specialists, in vivo performance of the operation and study of available literature. The analysis of two-dimensional illustrations for three-dimensional mental reconstruction is a time-consuming and difficult task. Courses for temporal bone dissection are infrequent and expensive. Cadavers are becoming increasingly hard to acquire, not all bone types are represented (due to variability in size and shape) and properties change shortly after the person dies (such as tissue deterioration). Condition of the bone, as well as age and race, will also

vary between specimens and cannot be fully represented by the small sample size within the laboratory. Facilities of this type are therefore costly and do not represent all possible scenarios that could be encountered in the operating suite. In effect, practical training for the ENT surgeon is restricted. Apart from observing or performing live implantation inside the operating suite, post-mortem evaluation remains the only practical alternative to in vivo middle ear dissection.

Due to the complex physiology of the middle and inner ear, there is significant risk associated with allowing inexperienced medical students or practitioners to perform the cochlear implantation. Blood vessels, nerves and muscles reside within the region and interference with these structures may cause irreversible damage. For example, harm to the facial nerve can cause facial palsy. Passive observation of an experienced surgeon, however, is inadequate for proper training of technique. Medical instruction has historically relied on a mentor/trainee approach, yet this has proven expensive, in terms of both time and money. A surgical simulator with force feedback for CI insertion would be a valuable supplement to current methods of specialist training.

1.6 Surgical Simulation

Haptic-rendered computer simulators with real-time control have increasing application in the area of medical education. Advances in computer processing power and the development of high fidelity force-feedback devices with specialised software enable the reproduction of realistic anatomical models that have real-world characteristics. Maximally invasive surgical procedures, such as cochlear implantation, can now be replicated in a hazard-free environment, allowing novice or skilled physicians to practice difficult or unfamiliar tasks with patient-specific models. Abnormal situations or complications may be replicated in a virtual environment where the patient is not put

at risk of injury. Although initial financial outlays are associated with establishing the simulator, ongoing costs of training may be minimised as opposed to those related to ongoing experienced instruction and the purchase of materials (such as the implants and cadaver specimens). Objective evaluation of surgeon technique is another potential benefit, offering a new level of ability assessment. A simulator could potentially provide the manufacturer with a model that can predict, display and track the dynamic behaviour of a prosthesis relative to operational influences such as electrode array location.

More recent simulations combine force feedback and visual representations with the objective of enhancing the degree of realism of the virtual experience. Haptic feedback is required in a surgical simulation to provide the user with a greater sense of immersion into his or her environment than would be expected from the visual representation alone. The user can not only see but also touch the virtual environment during model manipulation. Touch sensation is a vital information channel in real-world situations. Information relating tool/object interactions is inferred by the surgeon during clinical intervention via the sense of touch. During the CI insertion, the implant disappears out of surgeon view soon after advancement past the cochleostomy site. The specialist relies on sense of touch for the rest of the insertion. Importantly, the otologist must relinquish pressure after the point of first resistance to avoid trauma in the Basal turn region [33, 37, 39]. Despite its small size and flexibility, the implant could possibly rupture or traumatise surrounding membrane, should force exertion be too great. Force feedback is required in this intricate procedure to minimise the risk of cochlea damage. Haptic rendering should accurately replicate real-world forces and torques associated with tool/object interactions during surgery.

Surgical simulators are not yet expected to replace in vivo training or in vitro practical interactions with cadavers. However, simulators may provide a valuable

supplement to existing training schemes due to their reproducibility, low cost and resource availability. In this work, the design and development of a surgical simulator with force feedback to train specialists in cochlear implantation is presented. Literature shows that such a simulator is the first of its kind. In the following section, the objectives of the work are discussed. The thesis scope is then presented. Related work is assessed in the next chapter, including surgical simulators that provide kinaesthetic sensation and preceding studies for virtual modelling of the human cochlea. The rest of the thesis is concerned with simulator development, including model construction, implementation and objective validation of the system.

1.7 Objectives of the Work

A clinically valid surgical simulator is produced in this work, to supplement current methods of training ENT surgeons to perform cochlear implantation. The simulator will provide the user with visual and force feedback during electrode insertion into a three-dimensional virtual model of the human ST. Visual and force feedback are to be relayed to the user throughout the insertion. The haptic representation is to be based on physical, real-world data. Force profiles obtained from the simulation are to be compared with those produced by experimental insertion studies, in order to validate the model.

The first stage in system design includes an analysis of the electrode designs and techniques available to the surgeon, for the purpose of replicating the procedure. The next phase in development involves the geometric modelling of the human cochlea, specifically the ST chamber within this structure. A three-dimensional replica of the human ST is produced from measured data, for the purpose of virtual cochlear implantation. This is compared with a model of the cochlea derived from CT. The geometric model requires anatomical accuracy for it to be used in a clinical training

environment. Visualisation of the cochlea follows, with varying degrees of scene magnification, rotation and translation offered to the user, as in the real procedure. Visual and force rendering during user interaction with the model is to be realized using specialized software and commercially available haptic interfacing. The user will be able to perform real-time implantation into the ST using a haptic device to manipulate the environment and deliver the forces associated with the insertion back to the user. Force, torque and position feedback will be displayed at the Graphical User Interface (GUI). This data is logged to an external text file to derive output force profiles for system validation. Insertion force studies that are performed using the Nucleus® 24 Contour™ electrode with the Standard Insertion Technique (SIT) are used to validate the output from the simulation. This system will provide an objective assessment of surgeon technique during electrode insertion into a model of the human ST and will be used to train specialists in this clinical procedure.

The system design should allow for variation in angle of approach, insertion speeds and individual differences in cochlear morphology. The latter will involve creation of a parametric, patient-specific model of the ST. Physical attributes obtained from measured data (such as the coefficient of friction between the implant and ST walls) should be included in the haptic representation of the model. The visual and physical behaviour of the simulation should mimic a real-life scenario. System performance should be assessed qualitatively and quantitatively via insertion force experiments.

The end result will provide considerable benefits to both novice and skilled surgeons. The proposed system will provide a safe, cost-effective training environment in which the patient is not at risk. A patient-specific model will allow for individual variances in cochlear morphology, proving beneficial for pre- and post- operative

planning. The system will offer quantitative evaluation of insertion forces, as well as real-time electrode insertion depth. The former has previously been a subjective evaluation. Information of this type will assist in improving surgeon technique and force delivery during an implantation. It may also lead to improvements in electrode array design.

The BM is also modelled as a first approximation, with membranous physical properties attributed to the haptic representation. The final model leaves scope to replicate the membranous CP as a whole as well as the SV as an alternative chamber for implantation. This system will be the first of its kind to offer real-time visual and haptic feedback during an insertion of a CI into an anatomically accurate model of the human ST. It also demonstrates potential for use in other medical techniques, including laparoscopies, biopsies, catheter insertions or further prosthetic implantations where visual feedback is precluded.

1.8 Thesis Scope

The stages of work comprising the thesis have been divided into 7 chapters, which are sequential in relation to simulator design, implementation and validation. In Chapter 1, an overview is presented which provides detail as to the purpose of the work, including major objectives and the scope of the research.

An extensive review of the published literature that relates to surgical simulator development and implementation is undertaken in Chapter 2. The focus of the work is directed towards CI insertion and its realisation in a virtual reality environment. As such, the stages for implementation are reviewed, with reference to existing applications. Visualisation of human anatomy is considered, including available imaging modalities, three-dimensional reconstructions using surface- and volume- based approaches, as well

as visual rendering of the human cochlea. Haptic rendering processes, system design and constraints, as well as methods for object/tool collision detection and response are analysed. Benefits of surgical simulators are identified. Applications of existing surgical simulators with force feedback are reviewed, including soft tissue manipulation, temporal bone dissection and device insertions. The significance and unique contribution of the work is presented, including the benefits that this type of simulator will provide.

Insertion force experiments are carried out to evaluate and quantify force delivery during implantation of electrode designs and administration techniques considered for development of the simulator. In this work, key factors contributing to force output during cochlear implant insertion are identified and assessed. Results from this work, including coefficient of friction measurements, are used in the design of the simulator and to validate its behaviour, including real-time force output. In Chapter 3, the experimental setup for measurement of insertion and frictional forces, methodology and results are provided, as well as a discussion of the work and relevant conclusions.

A geometric model of the human cochlea is produced for use in the simulation. Its construction and visualisation is explored in Chapter 4. First, an insight into existing reconstructions and measurements of the delicate structure are documented. In the work, a model is produced from spiral CT and reconstructed using the imaging processing and analysis package Analyze (AnalyzeDirect, Inc.). Model shortcomings are discussed. Three-dimensional reconstruction of the human ST from measured data produces an anatomically accurate model for virtual CI insertion. Model realisation and validation are performed using ANSYS (Leap Australia).

The work in Chapter 5 focuses on simulator implementation, including model optimisation and visualisation in the Reachin API (ReachinTechnologies AB). The

scene is made dynamic through visual and haptic rendering, the processes of which are given in detail. Realisation of the interactive operation associated with insertion of a virtual implant into the ST cavity is discussed, including algorithm design and layout, system features and limitations. Haptic rendering of the scene requires sub-sampling of the carrier for real-time control. Specialised force algorithms are introduced to replicate the insertion force delivery associated with a real-world insertion.

Force profile results and electrode positions obtained experimentally in Chapter 3 are used in Chapter 6 to validate the simulation. Output force and position information is captured during an insertion performed by the user in the virtual surgical suite. These results are then compared with those obtained experimentally from the insertion force studies. A statistical analysis of the data is carried out on splines fitted to the output force profiles using TableCurve2D (Systat) and Excel (Microsoft), to determine the degree of similarity between results. Subsequent results and a final analysis are presented.

Major conclusions from the research carried out in this thesis are given in Chapter 7. This includes an analysis of the major contributions of the work and its limitations, as well as recommendations for future work and expansion of the project.

Chapter 2

BACKGROUND

2.1 Introduction

Three-dimensional visualisation and interactive manipulation of human anatomy in a virtual reality environment has attracted widespread interest from researchers around the world. Methods for the acquisition, registration and reconstruction of complex internal structures such as the middle and inner ear have advanced with imaging techniques, computer graphics and hardware capabilities, over the last few decades. Computer scientists, engineers and surgeons have recently focused on improving real-time interactions with the models, particularly in the area of haptic rendering. Research in this area is currently focused on producing a surgical simulator that offers the user realistic, real-time visual and kinaesthetic sensation in a fully immersive yet synthetic environment.

The aim of this chapter is to provide the reader with a background study of the work that has been done in this field of research. Visual and haptic rendering of an object is a vast area of study. The focus of the work is on the development of a three-dimensional model of the human cochlea with real-time visual and force feedback during insertion of an electrode array. In effect, the related work associated with visualisation of the temporal bone anatomy, specifically the cochlea, and haptic modelling for surgical simulation, will be examined.

First, the pipeline for anatomical visualisation, consisting of medical image acquisition, registration and visualisation is presented. Different schemes for topographic rendering of the temporal bone structures are discussed, including advantages and disadvantages for surface- and volume- rendering techniques. Methods for cochlear reconstruction are assessed; however geometric modelling of this structure is detailed in Chapter 4. A brief insight is given into animations of surgical procedures, including their applications, such as virtual CI insertion and the approaches that are adopted. The concept of haptic rendering, including collision detection and response, is introduced. With the addition of haptic feedback, the designer must also consider system constraints as well as the benefits of providing this utility. Existing surgical simulators that complement visual with force feedback are examined at length. These are clustered into applications including temporal bone dissection, soft tissue manipulation and device insertions. Examination of current simulators reveals the significance and unique contributions of the thesis, which are outlined at the end of this chapter.

2.2 Anatomical Visualisation Pipeline

The process of anatomical visualisation commences with either measured data or a medical image sequence. Prior to geometric modelling, the latter must first be registered or imported into a software program for analysis and processing. Important anatomical structures are identified in a process termed *segmentation* and these features are extracted from each image. Volume visualisation is realised by interpolation between image slices and rendering of the data. Either a surface-based or volume-based approach can be used to visualise the anatomical structure of interest. Illumination and shading effects may be added to the model to enhance its visual appearance. A pipeline for

anatomical visualisation is shown in Figure 2.1 and is described in the following literature.

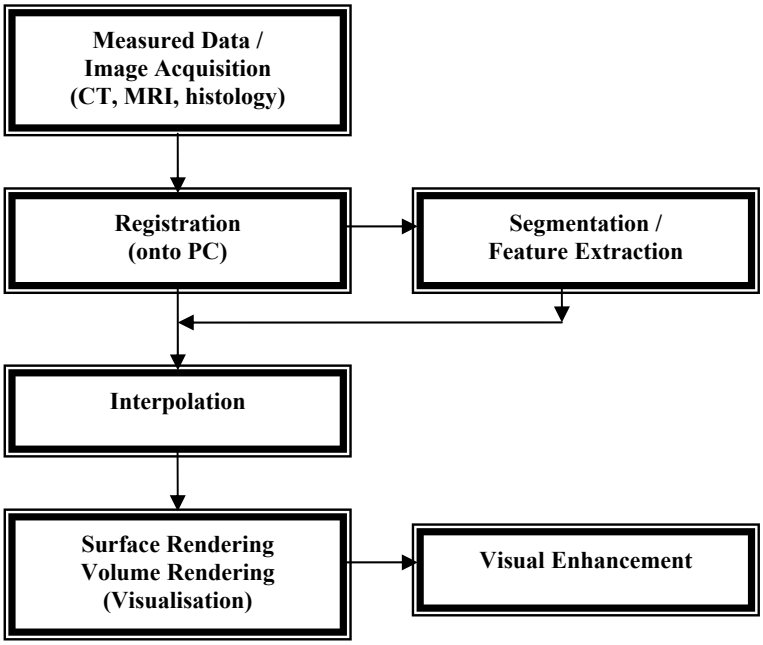

Figure 2.1. The Visualisation Pipeline.

2.2.1 Image Acquisition and Registration

In the following sections, the work is primarily concerned with visualisation of the temporal bone region. For this application, simulator development begins with the acquisition and registration of either geometric data or a medical image sequence. Imaging modalities including histology [6, 9, 10, 50-67], Computed Tomography (CT) [3-5, 7-9, 52, 55, 68-81] and Magnetic Resonance Imaging (MRI) [4, 5, 61-63, 68, 78, 81-84] have been popularly used to capture middle and inner ear data. CT and MRI are non-intrusive techniques, whilst histological examination requires cadaver dissection to acquire internal information. Computer-aided reconstruction of the temporal bone region has been implemented using these approaches.

2.2.1.1 Histological Sectioning

Histological sectioning is the process associated with viewing small anatomical structures such as cells and tissues for scientific evaluation via layered dissection [63]. To identify one structure from the next within a particular section, a somewhat standardised procedure is implemented for sample preparation. This involves fixation, decalcification, embedding, sectioning, staining and mounting [63]. Following material harvest, tissues within the cadaver are fixed to prevent movement or loss. Decalcifying fluid is then used to dissolve inorganic salts and to enable slicing [63]. The sample is then embedded with a sliceable substance to replace the decalcifying liquid. The final step in sample preparation is to dry, stain, mount and cover it with glass [63]. For applications such as computer-aided reconstruction of the middle and inner ear anatomy, further steps include sample registration on a computer and storage of this data in memory.

First, temporal bone cadavers are harvested post-mortem for histological sectioning. These form the basis for many three-dimensional reconstructions of anatomy within this region [6, 9, 10, 50-67], including the cochlea [6, 9, 10, 50-53, 55-58, 60, 64, 65, 67], and have been obtained from human [6, 9, 10, 50-54, 57-61, 65, 67] and other species such as cats [62], bats [63], dolphins [55] and guinea pigs [56, 64]. Following bone acquisition, treatment may include trimming [52, 64] and/or immersion in formol-saline [56, 64] and rinsing with water [9], or fixation with formalin [63]. Decalcification of the specimen can be performed with part ethylenediaminetetra-acetic acid in neutral buffered formalin [9, 52], or a formalin based solution [55]. Other sources use a 5% trichloroacetic acid solution, neutralisation and dehydration in graded alcohol solutions [63] for this process. Spurr's resin [9, 52], paraffin [55] and celloidin [10, 55, 59, 60, 63] have been used as embedding substances, in preparation for rigid sectioning. Prior to

sectioning, realignment holes may be drilled [51, 54, 55, 59, 63] and/or dye injected [10] into the prepared blocks for correct alignment and orientation during volume reconstruction. It is recommended that for accurate realignment, fiducial marks be employed [50, 59, 60]. Slice number, thickness and sampling frequency can vary. The number of slices can vary between 55 [58] and 630 [50, 65]. Sample thickness also varies (between 20pm and 200μm [58]), however depths of 20μm [10, 50, 55, 59, 63, 65] or 30μm [10, 53, 57, 62] are common. Sectioning may also be done horizontally [60, 61] and/or vertically [57, 61]. Staining is commonly performed with hematoxylin and eosin [9, 52, 57, 59, 60] and the specimens mounted in glass [59].

For image registration, the samples must be registered on a computer. Temporal bone sections may be digitised using a video camera that is connected to a computer with a frame-grabber card to capture the data as a sequence of high resolution images [9, 52, 58, 60, 61]. Microscopes can aid in the data acquisition process [60, 63] and still image cameras [51, 59, 62-64, 67] and/or a scanner [50, 54, 59, 67] are used. Structures within the sections that are mounted on microscopic slides may be manually traced onto white paper [10, 53, 57], with the aid of a slide projector. The data are then entered into the computer to produce three-dimensional coordinates. Digitiser tablets are also employed, where structure contours are traced onto the tablet (thickness may be recorded) [10, 53]. Voie et al. [56] acquire the data via laser-excited fluorescence optical sectioning, with scans extending from ST base to apex. Post-processing may involve image scaling [61] or changes in appearance (including transparency or colour [50]). Image sequences are stored in computer memory [50, 60, 64]. For image alignment, Adobe Photoshop has been used [51] for manual comparison, translation and rotation of adjacent images using superimposition [50]. Figure 2.2 is a digitised histological section with structures

differentiated by use of colour variation [63]; connective tissues are pink/red, bone and cartilage are blue/purple.

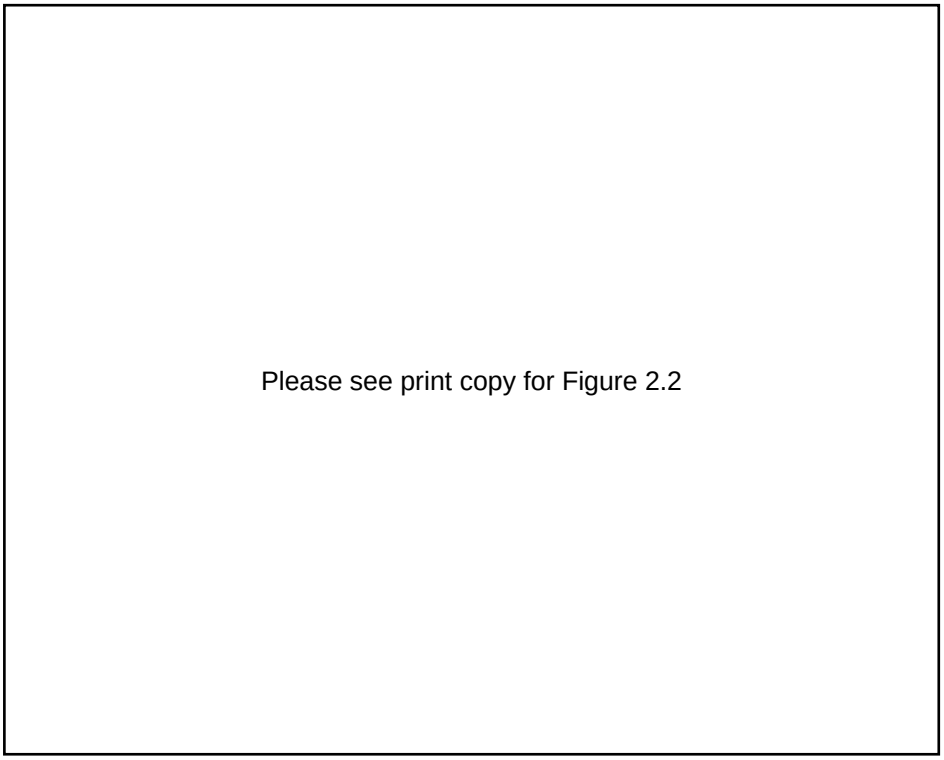

Figure 2.2. A digitised histological section of the ear of a moustached bat [63].

Histological sectioning can lead to the generation of high fidelity image slices representing the internal structures, due to the ability to produce thin sections [10]. This technique has facilitated the successful reconstruction of temporal bone anatomy, for visualisation, understanding relationships between structures [57] and for measurement of data [10]. Lutz et al. [57] maintain that any structure that is identifiable on a histologic section can be accurately reconstructed using computer technology [57]. It is recognised that histology complements other imaging modalities for temporal bone visualisation, such as Magnetic Resonance Microscopy (MRM) [63] and MRI [61], particularly for structure location and visualisation.

Despite these benefits, the most notable disadvantage in using histological sectioning for three-dimensional reconstruction of the middle and inner ear is that the donor must be deceased and the cadaver acquired in order to obtain the information. Further, errors in bone reconstruction may occur due to image misalignment [60, 63] or specimen shrinkage during its embedding [9]. The bones may also contain artifacts [50]. These defects can include stretching, tearing [75], folding [63], waving and variations in thickness or cutting angle [10]. The process associated with histological sectioning, including the stages of material preparation and image acquisition, is time consuming and comprised of many dedicated steps [10], which is in contrast to other imaging modalities such as spiral CT [58]. A lot of the work in this process is manual [10] and not only does this contribute to time and money spent on the procedure, but handling and preparation of the specimens can induce damage to the delicate structures that reside within the temporal bone cadaver [37].

2.2.1.2 Computed Tomography (CT)

CT is commonly used for pre- and post- operative evaluation of CI recipients [4, 8, 68, 70, 72, 73, 80, 85], for planning and assessment. It is used to determine any contraindications which may prevent implantation [4, 72], CI positioning and insertion depth [3, 4, 8, 52, 68, 73, 79, 85, 86], and for the purposes of anatomical measurement [3, 8, 52, 55, 73, 75] and modelling [3, 5, 7, 8, 52, 55, 69-71, 73, 75, 85]. CT is also an aid for medical instruction, to assist student and surgeon spatial reasoning [68, 70]. It is the modality of choice for post-operative imaging of CI patients [70], since MRI is a contraindication for cochlear implantation [72]. Unlike histological sectioning, CT is a non-invasive technique that can be performed on live subjects [3, 8, 52, 55, 68, 70, 71, 73, 76, 85], although in vitro imaging of cadavers is also done [75]. This imaging

modality can capture the intricate structures within the temporal bone region for virtual reconstruction. A number of sources have utilised conventional x-ray [75], post-operative high resolution radiography [52] or high resolution spiral CT [3, 5, 7, 8, 68-70, 72, 73, 85] for three-dimensional modelling of the temporal bone anatomy [5, 7, 8, 52, 55, 68-73, 75-78], including the cochlea [5, 7, 8, 55, 68-70, 72, 73, 76] and the electrode array [3, 4, 8, 69-73, 76, 79, 85].

Conventional radiography and spiral CT techniques work by measuring the degree to which tissues absorb x-rays [55]. The final result of the imaging is a sequential stack of two-dimensional images (Figure 2.3) of the temporal bone volume [75]. Image resolution will vary, depending on the image display and scanner resolution, as well as the size of computer memory for image storage [55]. Typical image intensity is 12-bit grey scale for spiral CT [69, 85], with a spatial resolution of $0.1mm^2$ pixel size [3, 68, 69, 73] for a 512 by 512 pixel matrix [55, 68, 85]. Inter-slice distance is generally 1mm [3, 5, 55, 68, 73], with a variable reconstruction (from 0.1mm [3, 69, 73, 85] to 0.5mm [68]) and number of images per volume (100 to 150 [3]). Temporal bone anatomy contains an array of fibrous tissues, bone and fluid, which vary between high and low densities. This scenario is well suited to CT since optimal imaging occurs where there is high contrast between tissue densities [55], particularly for bone which is of high density.

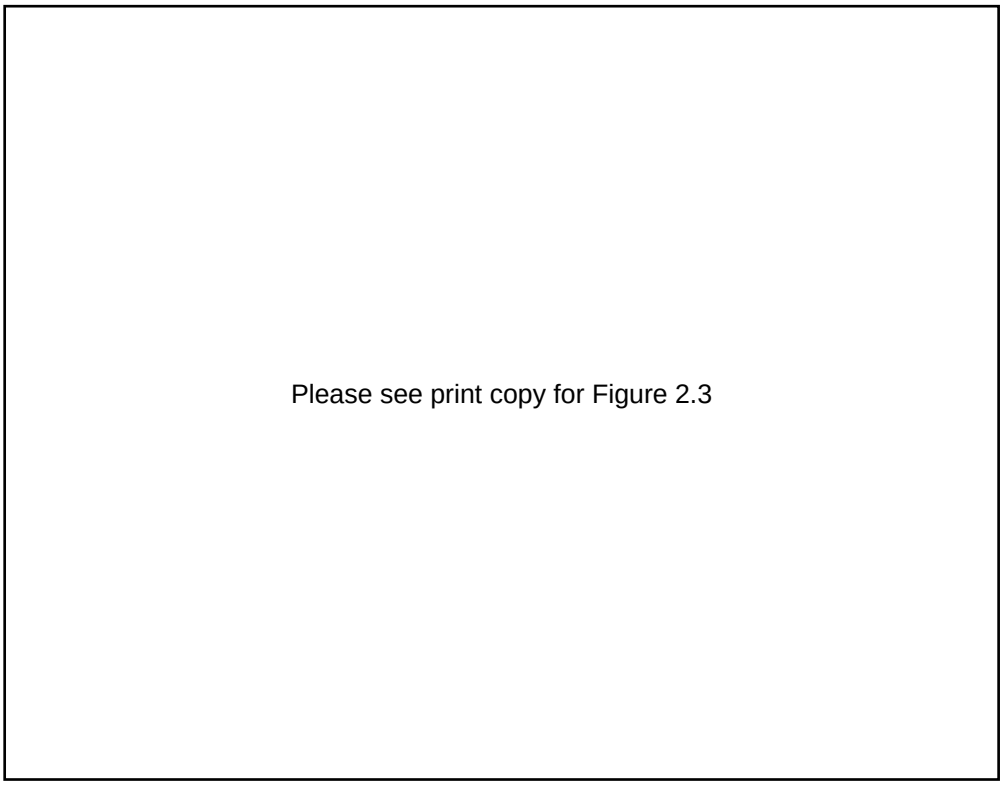

Figure 2.3. Two-dimensional CT image of middle and inner ear structures [70].

In comparison to histological sectioning, the time taken for sample preparation and scanning is minimal using CT [75]. High resolution, well aligned images of the cochlea and implant can be acquired [8] and imaging may be performed in vivo [8]. Despite the low cost and availability of traditional two-dimensional X-ray scanners or plain radiographs [52], off-axis projections can occur. Spiral CT provides cochlear height information [73, 85] and better, sub-millimetre resolution [8, 73, 85], yet not all structures can be precisely extracted from these images. Spiral CT offers overall superior image fidelity compared with the conventional radiograph [72, 73], particularly for thin slices and when spatial resolution is maximal [70]. CT can effectively capture data including structure size and condition, which is valuable for CI assessment [4].

Whilst CT is purposeful for temporal bone visualisation, the imaging technique has some drawbacks. Partial volume effects have been noted [52, 73, 76, 85] due to the

limited resolution of the scan [7, 8] and results in the averaging of border pixels with different intensity values [69]. The ST, SV and SM are not well delineated in CT images, once processing techniques such as thresholding have been applied. It is noted that spiral CT image deblurring may improve the image resolution, for clearer recognition of these important structures [69, 80, 87], particularly for representation of the ST which is the primary passage for CI insertion. CT has been successfully used to reconstruct high density structures such as the ossicles [4, 68, 71, 72, 75, 76], stapes [55, 70], bony labyrinth [4, 5, 68, 77] with semicircular canals [52, 68-70, 72, 73, 76] and the cochlear canal periphery (otic capsule) [3, 7, 69, 71, 72, 76]. However, soft tissue structures, including tendons, muscles [75], the BM [8], the Organ of Corti (OC) and Reissner's membrane [3] cannot be clearly represented in CT as opposed to MRI. Consequently, precise structural measurement [3], spatial orientation (such as defining the position of the CP within the cochlear labyrinth) and reconstruction of the cochlea is difficult [7]. Further, CT can miss abnormalities, including fibrosis and obliteration of the cochlea [4].

2.2.1.3 Magnetic Resonance Imaging (MRI)

MRI is a non-destructive medical imaging technique that uses nuclear magnetic resonance for the imaging of proton densities within the body [63]. MRM is similar to MRI, but with better spatial resolution for microscopic imaging of tiny structures [63]. MRI and MRM modalities are less commonly used than CT or histology for feature extraction of temporal bone anatomy, particularly for the purpose of pre-operative CI planning [5]. However, it is becoming more widely used for this application [82]. Reasons for the preference of CT include a higher cost for MRI [4, 5] and the difference in the type of features that need to be examined. MRI is more suited to soft tissue recognition [63] (including the cochlear nerve) than bone [82] (such as poor definition

of the mastoid quadrant [4]) and cannot be performed post-operatively on CI patients [4, 72]. It has been shown that MRI can reveal abnormalities [5], such as fibrosis [82], and cochlear congenital defects, where CT cannot [4]. The separate fluid-filled cochlear canals (ST, SV) can also be distinguished with MRI [5, 82].

Imaging and virtual reconstructions of temporal bone anatomy have been performed using MRI [5]. Hans et al. [5] use high resolution T2-weighted MRI to produce three-dimensional reconstructions of the SV and ST as separate chambers, notably in the Basal turn region. Images are acquired on a 1.5-T MRI scanner and slice thickness is 3mm [5]. The authors describe the quality of the three-dimensional renderings derived from the MR data as excellent [5]. Three-dimensional reconstructions of the ST, SV and SM fluid-filled cavities are produced by MRM, for individual chamber measurement, including cross-section area, length and volume [83].

Warrick and Funnell [62] produce surface reconstructions of middle ear anatomy using high resolution (0.13mm side voxels: 256 by 187 by 180 in total) MRI for the purpose of medical education. The tympanic membrane, malleus, incus and stapes were visualised in the work, while smaller structures were not able to be identified due to the limited resolution of MRI [62]. Virtual Reality Modelling Language (VRML) representations of inner ear anatomy, including the cochlea and semicircular canals, are produced from high resolution MRI [82]. The authors use 1.5-T MRI for pre-operative evaluation of CI candidates to identify cochlear abnormalities, to assist the surgeon in selecting the appropriate device and insertion procedure, as well as the preferred choice of ear (left or right) for insertion [82]. Boor et al. [68] use high resolution MRI for three-dimensional rendering of the cochlear nerves and labyrinthine artery. Daniel et al. [61] use high resolution MRM to model the mechanics of the human middle ear, including the ossicles, ligaments and muscles that reside within this region. MRM is

also used to produce a Finite Element Model (FEM) of the moustached bat, for the purpose of sound conduction synthesis [63]. An MRM slice used in the work is shown in Figure 2.4 [63]. A limitation of MRI is magnetic susceptibility, where there is signal loss at fluid/bone junctions [5].

It is suggested that multi-modal schemes, comprising data from more than one imaging modality, can enhance visualisation of temporal bone anatomy [69-71, 75, 81, 82]. This has been implemented for MRI in combination with CT [74, 81, 88], as well as with histology [61, 63]. To achieve this, co-registration of the images is required [88]. As well as cost implications [5], Hans et al. [82] claim that there are also practical limitations, with some institutions not wanting CT and MRI scanners in close proximity to each other.

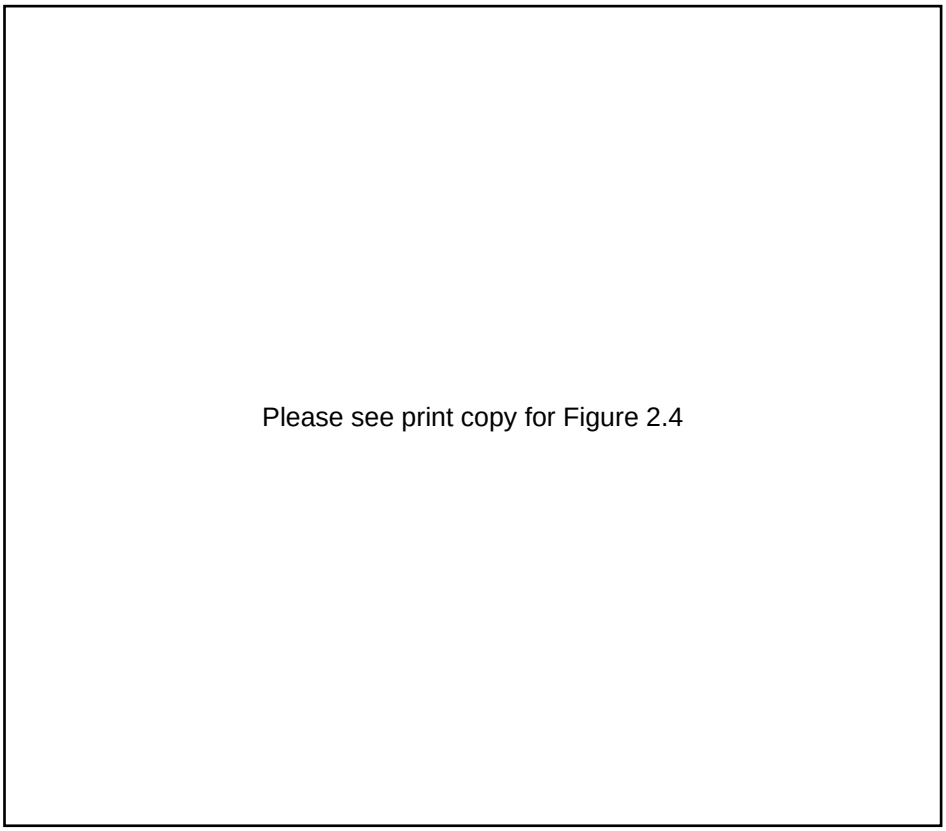

Figure 2.4. MRM of the temporal bone region [63].

2.2.2 Segmentation

In a process that follows data acquisition and registration, anatomical structures within the temporal bone are separated from surrounding parenchyma via *segmentation*. Structures within this region are identified and segmented using either a manual [9, 10, 50-53, 57-63, 65, 71, 75, 76], semi-automatic [5, 7, 62, 67-72, 76, 82] or fully automated approach [52, 76]. Entirely automatic segmentation is desirable [69], to reduce time, cost and manual labour, however an approach resulting in highly accurate feature delineation is difficult to achieve [62]. Often, complex structures can only be precisely defined using manual segmentation, however this is usually time-consuming and laborious [62, 63, 69, 75]. Manual and semi-automatic image processing methods are the most common for temporal bone reconstructions. This is mainly due to the vast number of complex shapes and variability of sizes of the structures within this region, as well as image artifacts [50, 65]. Manual techniques involve the user tracing structures, such as the cochlear boundaries [57], by hand (cursor) [9, 10, 50-53, 57-63, 65, 76] or cropping of images to include only relevant information [5, 82]. Regions of interest (ROIs) can be colour-coded for structure differentiation [4, 10, 50, 59, 61, 65, 70, 71]. Interactive, semi-automatic segmentation is often performed by the user selecting a threshold value based on pixel intensities [7, 62, 68-72, 82]. Recursive connectivity algorithms are also used, where a seed value is selected on the ROI surface [82] and areas not located on the surface are discarded. Region-growing algorithms are used [7, 69, 70] and artifacts are manually removed [70]. Erosion and dilation iterations are also performed [71]. Seldon [76] enables the user to select an automated, semi-automatic or manual edge-detection algorithm for ROI extraction. Xu et al. [52] provide a manual or automatic approach. The latter uses grey-level edge-detection.

Software packages that facilitate feature extraction from CT, MRI or digitised histological sections include SolidWorks (SolidWorks, Inc.) [51], OpenGL (Silicon Graphics, Inc.), Adobe Photoshop (Adobe) [50, 65], SURFdriver (University of Hawaii USA, University of Alberta CA) [75], the Application Visualization System (AVS 5, AVS/Uniras, Copenhagen, Denmark) [5] and Analyze [7, 69]. Custom programs, such as Fie (Funnell 2002) [61, 63, 75], 3D Slicer (Surgical Planning Laboratory, Brigham and Women's Hospital, Boston, USA) [71] and the Endovascular Surgery Planner (ESP) [82], have been developed for this purpose. An example of segmented temporal bone data, produced from a two-dimensional axial CT image, is shown in Figure 2.5. The separate structures of the ossicles, inner ear and nerves residing within the temporal bone region are clearly differentiated by application of colour-coding.

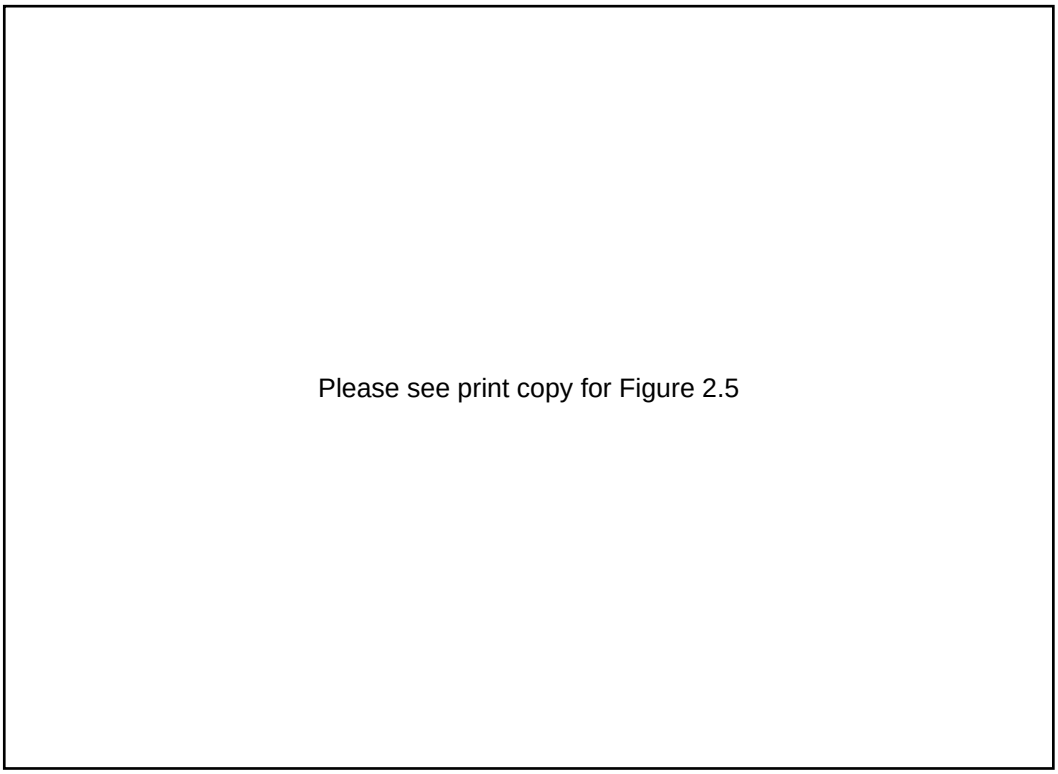

Figure 2.5. Axial image of the temporal bone and inner ear with segmented structures differentiated by colour [71]. Images were acquired from multi-slice CT.

2.2.3 Three-Dimensional Reconstructions of Temporal Bone Anatomy

Visualisation of the human anatomy in three dimensions basically fits into two categories: surface-rendering and volume-rendering [5]. This section of work describes the methods, differences, benefits and disadvantages for each of these, with an emphasis on the most common approaches implemented for display of the temporal bone and cochlea.

2.2.3.1 Surface-Rendering

Surface-rendering is the process of defining the periphery of an object and illustrating the topographical information as a two-dimensional depiction on the computer screen. Visual effects such as shading, illumination, transparency, colour and texture are added to the defined surface to enhance its realism. The resolved surface is comprised of geometric *primitives*, including lines, points and polygons of varying size and shape [89]. These primitives combine to represent a three-dimensional reconstruction of the boundary of an object within the virtual scene.

Earlier surface-based models simply used co-ordinates [9], splines [53, 57] or wire-frame [10] approximations to represent the boundaries of the cochlea in three dimensions. More recent approaches interpolate between segmented data or contours that form the structure's edge and this topology is converted into a polygonal mesh to define the surfaces of temporal bone anatomy [6, 7, 50, 51, 54, 56, 59, 61-65, 69-71, 75, 82], specifically the cochlea [5-7, 50, 56, 64, 65, 69-71, 82]. Commercial programs including Analyze [7, 69], SURFdriver (SURFdriver Software) [75], the Application Visualization System (AVS) (Advanced Visual Systems, Inc.) [5] and 3D Studio Max® (Kinetix) [51, 54, 69], as well as locally developed software such as Tr3 [61, 63, 75], are used for surface reconstruction of the anatomy [7, 61, 69] and also the CI geometry [6, 69]. The marching cubes algorithm (with triangular polygons), developed by Lorensen and Cline

[90], has been popularly implemented for surface reconstructions [5, 50, 62, 65, 67, 71, 82]. The dividing cubes algorithm (with points instead of triangles), which was originally developed by the same authors [90], has also been used to create the cochlear model [76]. The marching cubes algorithm has been implemented using the Visualisation Toolkit (VTK) software (open source) for geometric renderings [50, 65, 71]. Three-dimensional surface models have been created from VRML [82], due to its portability [62] and cross-platform functionality [82], usually as either Indexed Face Sets (IFS) [69] or Indexed Triangle Sets (ITS) [62, 63]. Givelberg et al. [91, 92] create surface renderings of the SV, ST and BM using computational grids, where the interconnecting points are modelled by elastic springs. Physical properties are added to the model [91], as performed in FEM of surface shells [6, 51, 54, 61, 63, 93], for dynamic mechanical modelling.

For meshed surface reconstructions, there is a trade-off between the number and/or size of the polygons comprising the surface (defining anatomical accuracy) and the display quality [61, 69]. Optimisation techniques are applied to reduce the number of polygons whilst maintaining sufficient topographical fidelity. Yoo et al. [69] employ an optimisation function within 3D Studio Max to iteratively reduce the number of polygons representing the cochlear shell. Texture mapping is applied to reduce the geometrical complexity of the electrode array itself [69]. Polygon decimation has been carried out in VTK [62] to reduce the total number of polygons and hence model complexity. Optimisations such as these lead to improvements in visual rendering performance, due to a decrease in file size [69] and therefore an increase in computational efficiency. Visual quality may be enhanced by the smoothing of edges [60] (decreasing the crease angle in IFS [69]), Gouraud [59] or Phong shading, as well as Shaded Surface Display (SSD) [4, 60, 70], and the addition of material properties (such as shininess) or lighting effects [62].

An example of a surface-rendered model of the temporal bone, created via the marching cubes algorithm, is shown in Figure 2.6.

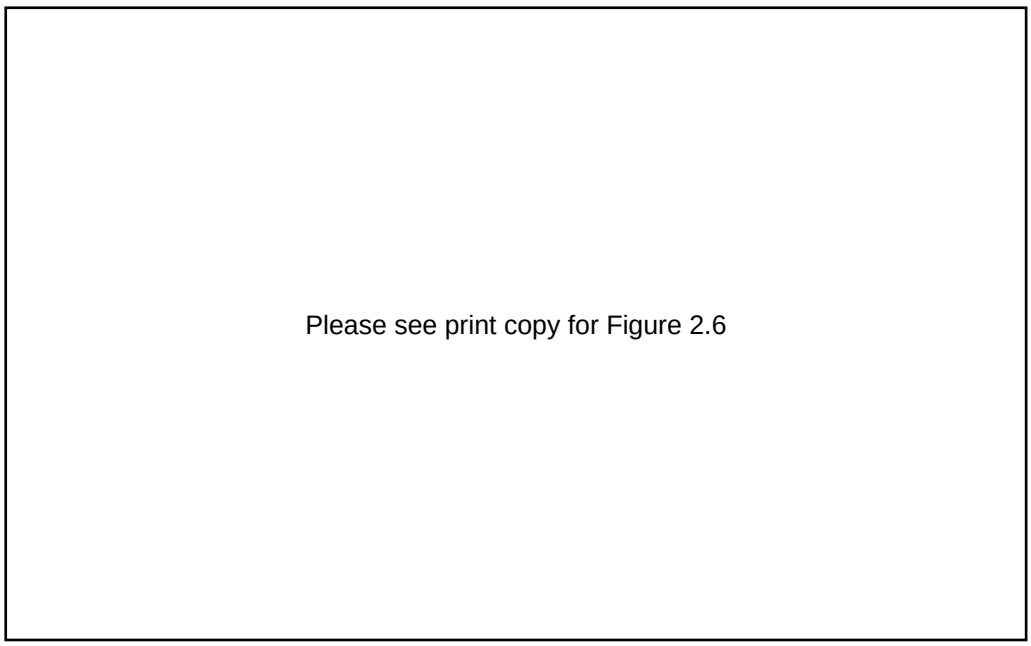

Figure 2.6. Surface-rendered image of the temporal bone and adjacent anatomy [71]. Renderings were done using the marching cube algorithm, implemented in VTK.

2.2.3.2 Volume-Rendering

Volume-rendering is performed on a three-dimensional dataset to represent the entire object as a two-dimensional image [89]. Unlike surface-rendering, volume-rendering is concerned with visualisation of the whole set of volume elements (voxels) that comprise the object and not just its surface. The fundamental difference between the two techniques is that surface-rendered objects contain no information below the surface of the object, whereas the internal components are represented by the volume-rendered object. Therefore the final image is a sub-sample of the entire volume, whereas the surface-based image represents only the outermost layer of the object. Ray-casting has been used for volume visualization in medical applications of this type [5, 68, 78, 94]. It is a process

where virtual rays are sent through the volume of voxels [89]. Samples of voxels that lie along these lines contribute to intensity and opacity, and are accumulated to produce a final value on the object's surface [89], which represents the internal volume data. Various types of ray-casting include back-to-front projection of the rays [94], front-to-back projection and maximum intensity projection (MIP) [4, 5]. For MIP, the highest voxel intensity is mapped to white and the remaining voxel intensities are relative to this value [5].

A ray-casting algorithm is implemented for display of the inner ear, including cochlear structures [68]. The results are combined with surface shading to produce a model for virtual endoscopy of the inner ear region, as shown in Figure 2.7 [68]. Ray-casting is also utilised by Pflesser et al. [78] for display of the temporal bone anatomy during a virtual mastoidectomy. Hans et al. [5] apply four different techniques for visualisation of the individual chambers of the ST and SV within the cochlea: MIP, ray-casting with either transparent or opaque voxels and a surface-rendering method that uses the marching cubes algorithm. Experienced, yet subjective evaluation of the results showed the latter two approaches to render the individual structures with greatest distinction, with assessor preference given to the isosurface renderings using MRI [5]. The authors [5] found the marching cubes algorithm 'simple and elegant' for isosurface visualisation, however it is sensitive to image noise and surface position is dependent on a user-selected threshold value [5]. It was determined that high quality renderings of the ST and SV are dependent on the imaging quality and post-processing methods [5]. Brown et al. [94] use a back-projection algorithm for display of the cochlear volume.

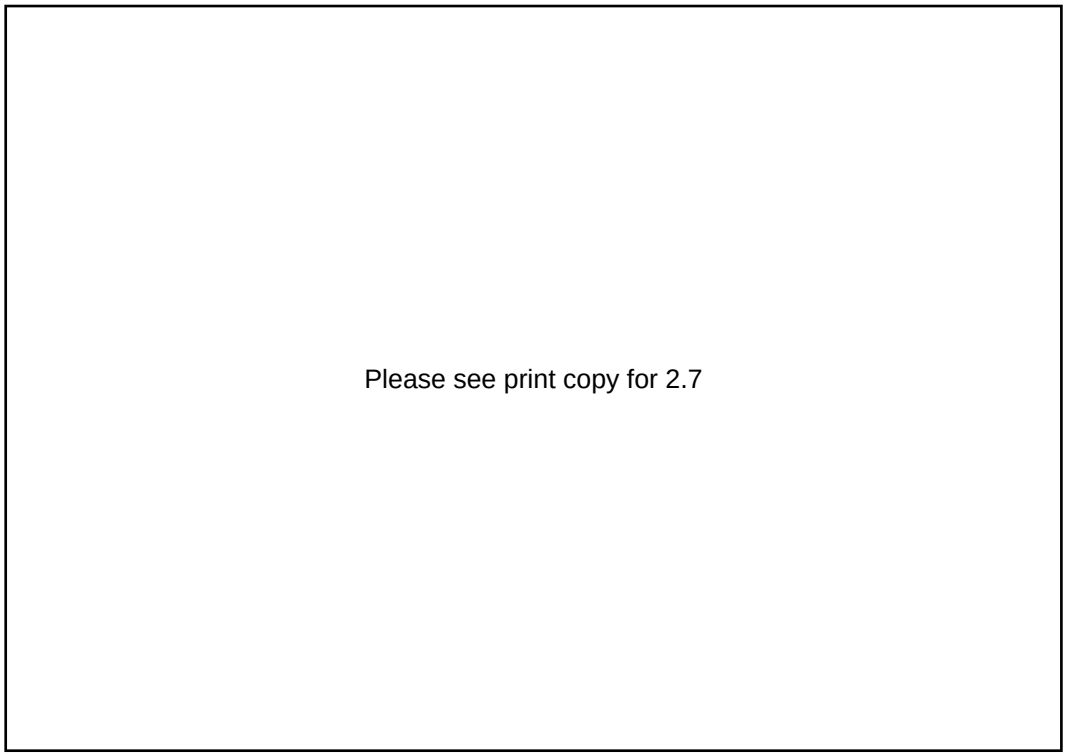

Figure 2.7. Two-dimensional image of a three-dimensional temporal bone volume projected via ray-casting [68].

Techniques other than ray-casting have been used for temporal bone visualisation. A hybrid approach is applied [70], where the authors combine surface- and volume-rendering for reconstruction of middle and inner ear structures (Figure 2.8). The temporal bone is visualised using a transparent volume rendering method, where voxel transparency is variable [70]. A FEM of the cochlea is produced in ANSYS [95], where each element represents a section of the cochlear volume (Figure 2.9). Material properties are assigned to the voxels to synthesise the behaviour of the cochlea during electrode stimulation [95]. Similarly, a FEM of the ST is produced in ANSYS to model a virtual CI insertion [41]. The helico-spiral approximation [7] is used to represent ST curvature and the physical properties of bone, BM and CI are included in the model [41]. The insertion

is simulated in ANSYS, for analysis of the array trajectory and contact pressures, as well as the effect of variation in electrode stiffness on force output [41].

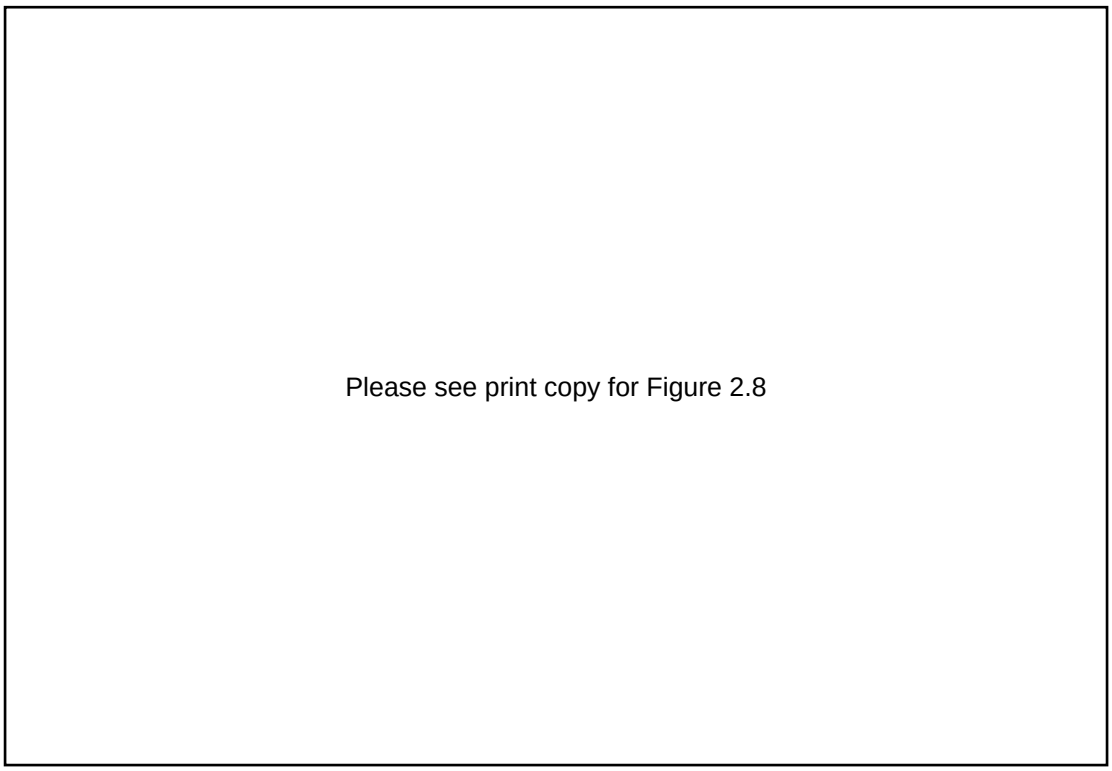

Figure 2.8. Volume-rendering of the temporal bone (white) with surface-rendering of the inner ear structures (coloured) [70].

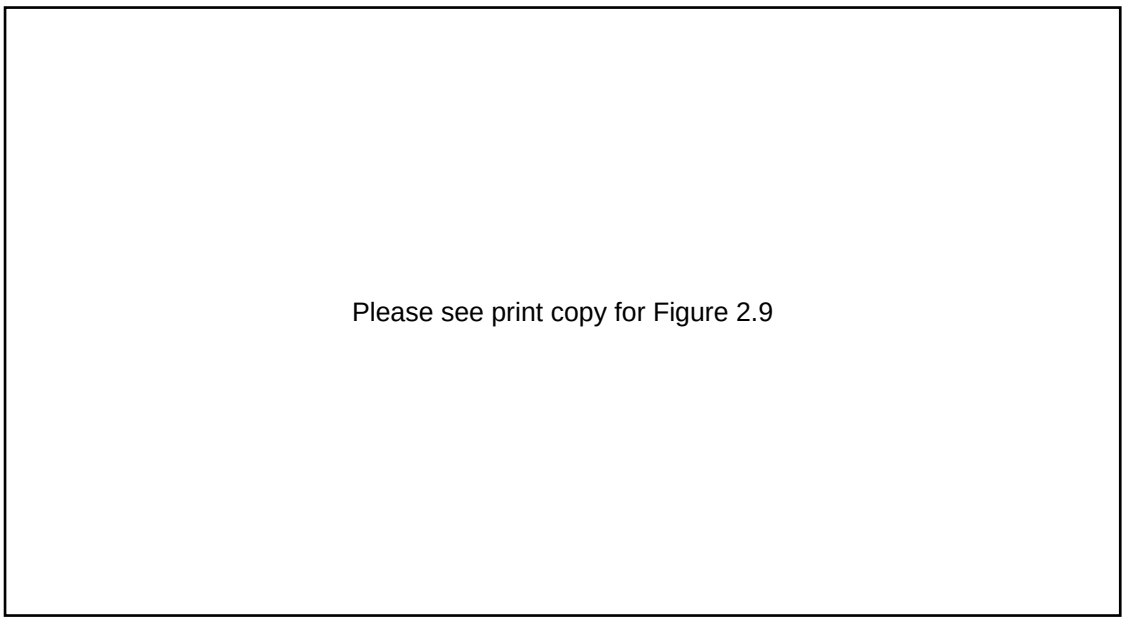

Figure 2.9. Finite Element (FE) volume model of the cochlea in ANSYS [95].

Volume-rendering is usually more computationally expensive than surface-rendering [75], since there is more data to process. However, the former does offer the advantage of providing intrinsic information, including internal structures, from within the volume as opposed to only representing the object boundary [70, 75, 82]. Decreamer et al. [75] suggest that volume-rendering produces a more realistic representation since voxel intensities are used in the projection. However, a subjective assessment of the temporal bone structures [5] has revealed a preference for surface-based rendering. Depending on the application, a surface- or volume- based approach can prove successful for accurately representing middle and inner ear structures.

2.3 Virtual Rendering of the Cochlea

In summary, the cochlea has been rendered as a virtual, three-dimensional depiction from geometric measurements [7, 26, 69, 91, 92, 95], histology [6, 9, 10, 50-53, 55-58, 60, 64, 65, 67], MRI [5, 82] and CT [5, 7, 8, 55, 68-70, 72, 73, 76]. Manual [9, 10, 50-53, 57, 58, 60, 65, 71, 76], semi-automatic [5, 7, 62, 67-72, 76, 82] and automatic [52, 76] techniques have been used to segment the structure from various imaging modalities. Three-dimensional reconstruction of the cochlear spiral has been successfully implemented using both surface-based [5-7, 9, 10, 50, 53, 56, 57, 64, 65, 67, 69-71, 76, 82] and volume-rendered [5, 68, 94, 95] approaches. Representation of the ST [56, 64] and SV partitioned as separate chambers has also been achieved [5, 82, 91, 92, 95]. Visualisation of the CI itself has been performed for post-operative assessment [3, 4, 8, 52, 69-73, 76, 79, 85], with limited sources providing animation of the actual insertion procedure [69]. Surgical animation of this type, and for similar applications, will be discussed in the next section.

2.4 Animation of Surgical Procedures involving the Cochlea

Three-dimensional modelling and visualisation of the human cochlea is useful for anatomical analysis. A medical student or practising otologist may gain further benefit from a simulation in which an operation on the inner ear structure can be synthesised. *Animation* adds this new level of dynamic movement to the scene, where investigative, exploratory or invasive procedures are recreated in a virtual environment.

Endoscopy is a medical procedure where internal organs are viewed using a fibre optic device (endoscope). This has been enacted for the virtual viewing of inner ear structures, including the cochlea [67, 68, 70, 71]. Boor et al. [68] use a volume-rendering approach, combining ray-casting with surface shading, to display the middle and inner ear structures. The authors [68] recreate a *virtual flight* through this region, from different viewing angles. Figure 2.10 shows one such scene, including the apical turn of the cochlea and modiolus, where the structures are derived from CT [68]. An endoscopic *fly-through* is also created for viewing middle and inner ear structures, including the cochlea [67]. A movie is produced of this scenario, from Adobe Premiere 5.5, as one progresses from the middle to inner ear regions. Two-dimensional CT images are superimposed on three-dimensional surface reconstructions, produced via the marching cubes algorithm in VTK, for endoscopic viewing of the temporal bone region, including the cochlea and an inserted implant [71]. A three-dimensional rendering of the implant during a virtual endoscopy (VE) is shown in Figure 2.11 [71]. The software '3D Slicer' facilitated real-time navigation of the anatomy, where the authors used a surface-based approach [71]. Seemann et al. [70] co-register three-dimensional structures produced from volume-rendering of the temporal bone with three-dimensional surface reconstructions of the inner ear anatomy. VE navigation of these structures is realised using dedicated software InvitoVR® which is operated by a user-controlled glove [70]. The inserted electrode

array and cochlea could be distinguished, yet soft tissue structures in this region were not identifiable [70].

Advantages of the VE application include the display of structures that the surgeon may see in the real procedure and may reveal regions that might be otherwise obscured by surrounding anatomy as well as their internal components [68]. Pre-operative evaluation by VE is an asset, for detection of abnormalities [68, 70] as well as the optimal approach for surgery [68, 70]. It is also useful for three-dimensional viewing of the complex anatomy of the temporal bone [70], particularly the cochlea, which assists the user in understanding the three-dimensional spatial relationships, as opposed to a mental recreation that is required by two-dimensional image sequences [70]. VE may be applied for pre-operative assessment of the cochlea for device implantation, including information relating to the best ear for implantation, any malformations or contraindications, as well as for post-operative evaluation of success [70] (such as the final position of the implant).

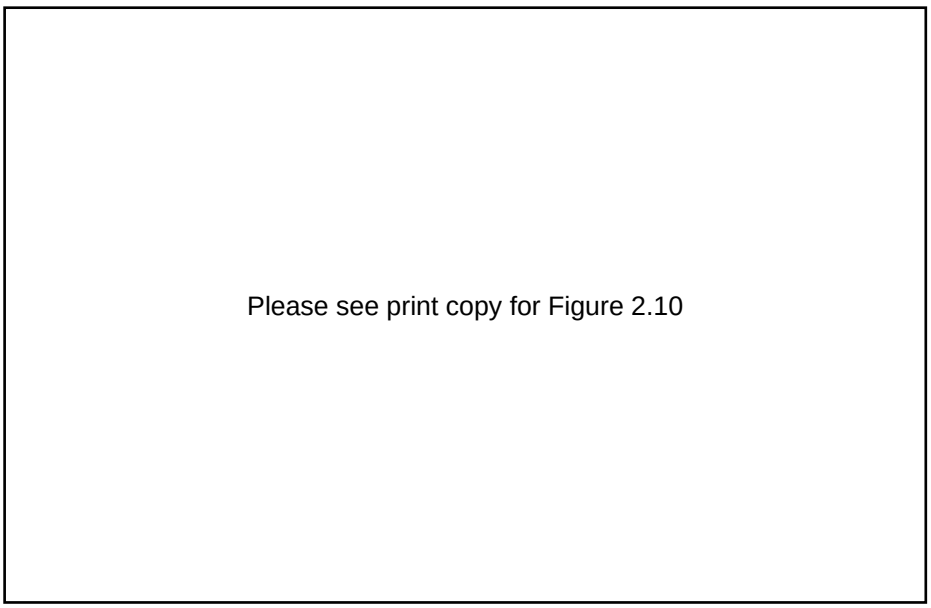

Figure 2.10. An endoscopic view of the cochlear apical turn and the modiolus [68].

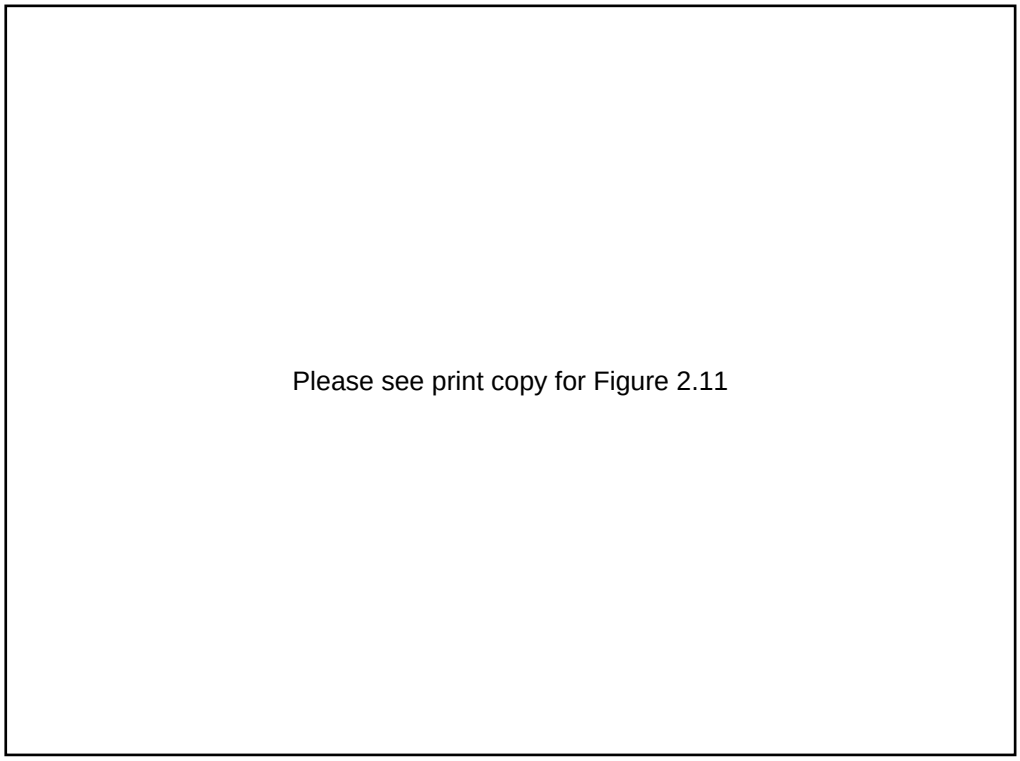

Figure 2.11. Virtual endoscopy: picture shows a CI and a two-dimensional image slice superimposed on the rendered scene [71].

Animated CI insertions are limited, however two-dimensional and three-dimensional models of the human cochlea have been developed for this purpose. Chen et al. [26] constructed a two-dimensional, FE replica of the ST to determine electrode positioning and force delivery during an insertion. In this application, geometric models of the ST and electrode array are synthesised from measured data. The mechanical properties, including array stiffness, are included in the design and the electrode behaves in accordance with the equations of motion as it deforms upon collision with the ST walls [26]. Whilst this is a good preliminary analysis of the effect of stiffness on insertion forces and analysis of the array trajectory, there are limitations of the model. The representation is only two-dimensional and does not allow for linear or twisting motion of the electrode in the other plane [26]. Further, the change in height (and hence tilt) of the cochlea is not considered (the authors simplify the analysis into two dimensions) [26].

One major drawback for this technique [26] is that the insertion is not performed in real-time: the FEM approach can prove quite slow. Further, the user does not interact with the electrode during the insertion, rather the computer simulation directs and monitors the implantation. A FEM of the ST is produced for CI insertion which is modelled in ANSYS [41]. Physical properties of the implant itself, including stiffness, the cochlea (bone and membrane characteristics) and interaction between the two (including coefficient of friction) are attributed [41]. Whilst the insertion does not occur in real-time, the three-dimensional analysis of the insertion process provides relevant information on electrode array trajectories and insertion forces in relation to stiffness properties of the electrode [41].

Surface representations of a real and idealised cochlea (based on Cohen's model of the central path traced by the cochlea) are produced for visualisation of CI insertion on the internet [69]. The polygonal models are visualised on the World Wide Web (WWW) using VRML 2.0/97 with Java code (Figure 2.12). VRML facilitates animation of the insertion and enables the user to select criteria for insertion [69]. This user interface offers the controller the following functionality: selection of electrode type, starting a single insertion loop or performing a step-wise motion of the electrode along the centre of the cochlea and changing the cochlear transparency [69]. Further, information regarding insertion depth, angle and individual electrode frequency is monitored [69].

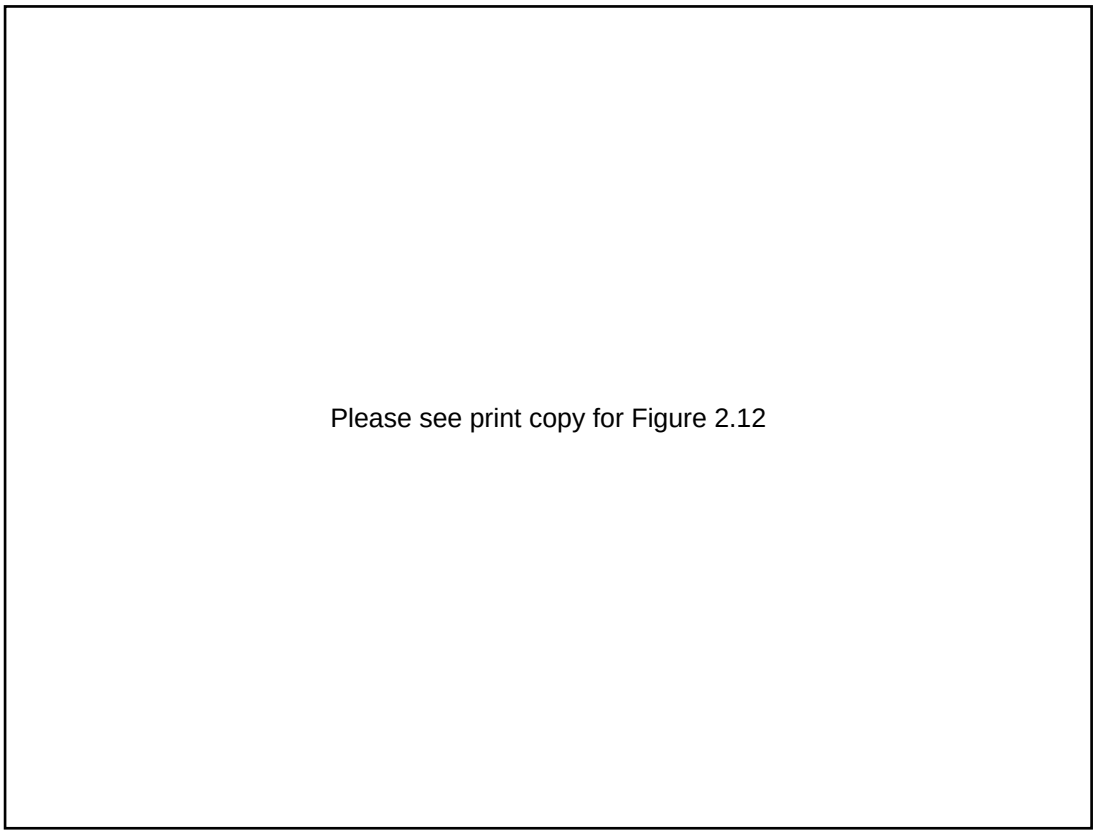

Figure 2.12. A transparent rendering of a CI insertion [69]. The Graphic User Interface (GUI) is shown in the figure. The CI is visible.

The simulation offers a three-dimensional perspective of cochlear implantation [69], however user interactivity is again limited and does not allow the user to have real-time control for arbitrary electrode positioning. There are no force input or feedback and the implant trajectory is pre-determined. Optimisation techniques were required to achieve maximum performance [69]. The models of both the real and idealised cochleae are inaccurately depicted as a single chamber and the authors concede that future developments should include rendering the ST, SV and SM as separate entities, as well as the electrode trajectory following the inside of the ST chamber instead of the central path of the cochlea [69]. Fully automatic segmentation of the structure is also desirable [69], to save time expenditure on manual intervention, as well as the representation of more

structures surrounding the cochlea by use of multiple imaging modalities (co-registering CT with MRI) [69]. However, the authors [69] have produced a valuable educational and research tool for this type of surgical procedure.

2.5 Haptic Rendering

Visual rendering of a scene captures induced changes in surface topology. The degree of realism of the virtual environment may be enhanced by the introduction of force feedback [96]. The integration of sight and touch sensation provides the user with a heightened sense of immersion than visual feedback alone [97]. In this section of work, the concept of haptic rendering within a closed loop control system is discussed. System design constraints are considered as are the benefits of a surgical simulator that offers force feedback. The rest of the section provides a detailed assessment of simulators of this type, in order to highlight the unique contribution of the research.

2.5.1 Haptic Feedback in Closed-Loop Control

Haptics is derived from the Greek word 'haptikos' meaning the ability to touch or grasp [98]. A haptics device is an interface between the computer and user that provides the medium for force delivery [99]. The force feedback is part of a closed loop system. First, the user exerts a force (either linear and/or twisting) by moving the arm of the haptics device. This causes proportional movement of the stylus within the virtual environment. Haptic rendering follows, where tool/object interactions are detected and force components are calculated during the collision. The model then reacts to the user-initiated manipulation, by moving or deforming accordingly. This process is referred to as *collision detection and response* [100, 101]. The force information is relayed back to the user through the haptic interface from which the motion was instigated [102]. This control

loop is summarised in Figure 2.13. During collision detection and response, topographical changes are relayed to the user through variation in object appearance at the GUI. The process of monitoring and applying appropriate changes in graphics is referred to as *visual rendering*. It is the process of determining the data to be displayed and the method used to display it. Illumination and shading models can be applied to the surfaces during this procedure, to facilitate visual recognition. Haptic rendering complements visualisation of anatomical structures for surgical simulators. However, there are design constraints that must be considered for stable, accurate modelling of organ behaviour in the virtual scene.

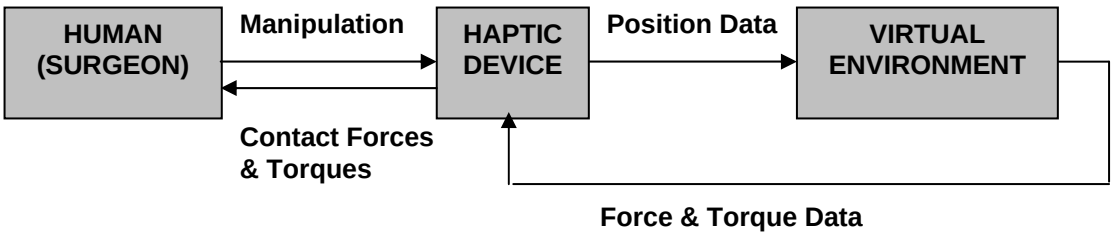

Figure 2.13. Haptic interface as part of a closed-loop control system.

2.5.2 System Design Constraints

The functionality of haptic-rendered surgical simulators is restricted due to limitations in the current technology. This includes finite medical image resolution, computer hardware limitations and haptic-loop update constraints. Addition of force-feedback often causes a trade-off between haptic algorithm complexity and graphic display quality, due to real-time constraints [100]. Until the recent increase in computer processing speeds and the development of specialised haptic hardware and software, these constraints have impeded the development of real-time haptic-rendered

simulations. Visual update rates can be as low as 20Hz to 30Hz [100, 103, 104], whilst forces must be calculated at a high frequency (at least 500Hz to 1000Hz [97, 100, 103-110]) to maintain haptic fidelity and hence system stability [97]. Computation of visual and force feedback must occur in real-time [101]. In effect, complex systems (with high spatial resolution and pixel intensity) are regularly simplified to reduce computational overhead [102]. Physical characteristics are added to anatomical models to synthesise real-world attributes [103]. The composition and behaviour of anatomical structures, such as the human middle and inner ear, are quite complex [111, 112]. The representations may, however, be approximated in the virtual environment so that interactive model manipulation can occur in real-time, without noticeable delay. However, the model must still remain visually realistic [113].

Introduction of torque further increases algorithm and computational complexity [109, 114]. Delays in the haptic loop can cause system instability, program termination or provide non-realistic feeling, with lack of precise control [103]. Force feedback projected in this way detracts the user from the current operation instead of providing benefit. Further, the initial price of simulator establishment is quite high, especially for systems incorporating a six-degree-of-freedom (6DOF) haptic device. Despite the high cost for system set-up, the hardware is low-cost (particularly the PC) relative to long-term expenses incurred by the student and sponsoring institution during the course of medical training [113, 115]. The inability to accurately measure force and torque contributions in vivo for some medical procedures, such as CI insertion, also inhibits accurate force modelling.

2.5.3 Benefits of a Haptic-Rendered Surgical Simulator

Haptic-rendered surgical simulators offer a new level of medical training. Dawson and Kaufman [113], and Basdogan et al. [112] provide extensive reviews of the benefits, as well as system constraints [112], for this type of application. The traditional mentor-trainee, see-do teaching philosophy may be supplemented by an interactive, cost-effective and, importantly, a safe virtual environment in which to practice difficult, new or abnormal procedures [113, 115]. Realistic, real-time display is essential and although there are challenges to overcome in this respect (as mentioned previously), a simulator can re-create patient-specific anatomy in a consistent, efficient manner [105, 113]. Once the system is constructed and validated, an objective measure of surgeon skill is possible [112, 113, 115]. The surgical simulator can be used for ongoing surgeon assessment, training and practice, for novice [105] and skilled physicians alike [113]. Pre-operative rehearsals of specific procedures can be carried out on patient-specific models without putting the patient at risk of injury [115], or the specialist at liability. It also removes any political or moral issues that might otherwise arise [105]. This type of training offers flexibility as well as reproducibility [115]. Since the surgeon acquires information through the sense of touch [112], particularly during cochlear implantation where the full insertion cannot be seen, force feedback from the simulation is essential.

2.6 Existing Surgical Simulators with Haptic Feedback

An extensive number of research and commercial groups have developed real-time, three-dimensional visualisations of surgical interventions, yet systems incorporating force feedback form a much smaller subset. This is due to the more recent origin of haptic interfaces and the supporting software. The overall concept of haptic rendering has been introduced, as well as the benefits it may provide for the purpose of surgical

simulation. In the following section of work, a detailed assessment of existing haptic-rendered simulators for medical education is given.

Surgical simulators with haptic feedback that will be examined in the work may be divided into three categories: bone drilling, soft tissue simulations and device insertions (which may be prosthetic implantations or exploratory surgery). Within these areas, virtual routines include temporal bone dissection [66, 74, 111, 116-131], needle insertion [132-144], for spine biopsy [133, 134], neurosurgery [139] and lower back puncturing [145], prostate cancer diagnosis [146] and treatment [137], catheter insertion [135, 136, 143, 144, 147, 148], human thigh [149] and knee [150, 151] examination, endoscopy [127, 128, 152-156], laparoscopy [157-163] for bile duct exploration [158], gall bladder removal [158, 161] and abdominal surgery [164] and soft tissue palpation [100, 146, 165-169]. Haptic-rendered surgical simulations may involve multi-point [74, 100, 111, 119, 120, 129] or single-point [100, 110, 165, 170, 171] collisions, proxy-based methods [111, 119, 120, 129] and surface-based [100, 111, 166] or volume-based [66, 117, 118, 123, 124, 133, 134, 143, 170, 172] haptic approaches. Haptic devices used in the simulations are either developed locally [162, 163, 168, 169, 173-176], usually designed for a specific application, or obtained commercially [66, 100, 111, 119, 121, 143, 157, 158, 177]. The most common of these devices are the Phantom Desktop (3DOF force output) [117, 118, 121, 123, 124, 131, 133-135, 139, 142, 144, 164-166, 170, 174, 178-180] and the Phantom Premium (6DOF force output) [137, 170] (SensAble Technologies). Software selection varies for visualisation and haptic algorithm implementation: the most common programs include the Reachin API (Reachin Technologies AB) [142, 178] or an OpenGL/GHOST® SDK (SensAble Technologies) approach [66, 117, 118, 139, 143-146, 157, 164, 165, 177], which offer basic collision detection and response algorithms.

2.6.1 Temporal Bone Dissection

Access to inner ear anatomy is acquired via a *mastoidectomy*: a process by which the mastoid quadrant of the temporal bone region of the middle ear is dissected via drilling. For this procedure, the surgeon employs drilling burrs which differ in size, shape, flute orientation and depth. Cutting burr motion is also arbitrary, although Agus et al. [124] describe it as a polishing motion. Whilst performing the mastoidectomy, the otologist uses irrigation to cool the bone (since drilling produces heat) and suctioning of the bone paste that forms as the bone dust combines with the coolant. Haptic-rendered simulations of this maximally invasive procedure have been constructed for surgeon training, with varying degrees of design specifications [66, 111].

A working prototype has been developed by Wiet et al. [66, 117, 118] for real-time manipulation and drilling of a volumetric model of the temporal bone. Visual, force and aural feedback are generated during the bone removal, irrigation and suctioning actions, to provide a life-like scenario of the true operation [66, 117, 118]. The authors use a direct volume-rendering approach, with three-dimensional texture mapping, to visually display the bone data in OpenGL [66, 118]. A spring-based model is applied for rendering of forces during burr-bone interactions; where a ‘virtual “force-field”’ [118] encircles the burr and force calculation is performed based on spring stiffness and burr penetration depth [66]. Forces are delivered back to the user through two 3DOF haptic devices [66, 117]. Sounds associated with the drilling and suctioning are synthesized by processing audio samples taken from an online library [66]. Fluid effects, including blood and water flow, are not included in the simulation [66]. Magnification is enabled [117], which the surgeon relies on during the real procedure, increasing the level of zoom upon approach to the cochlea. Burr properties, including

size and shape, are selectable [66]. An illustration of this system is provided in Figure 2.14.

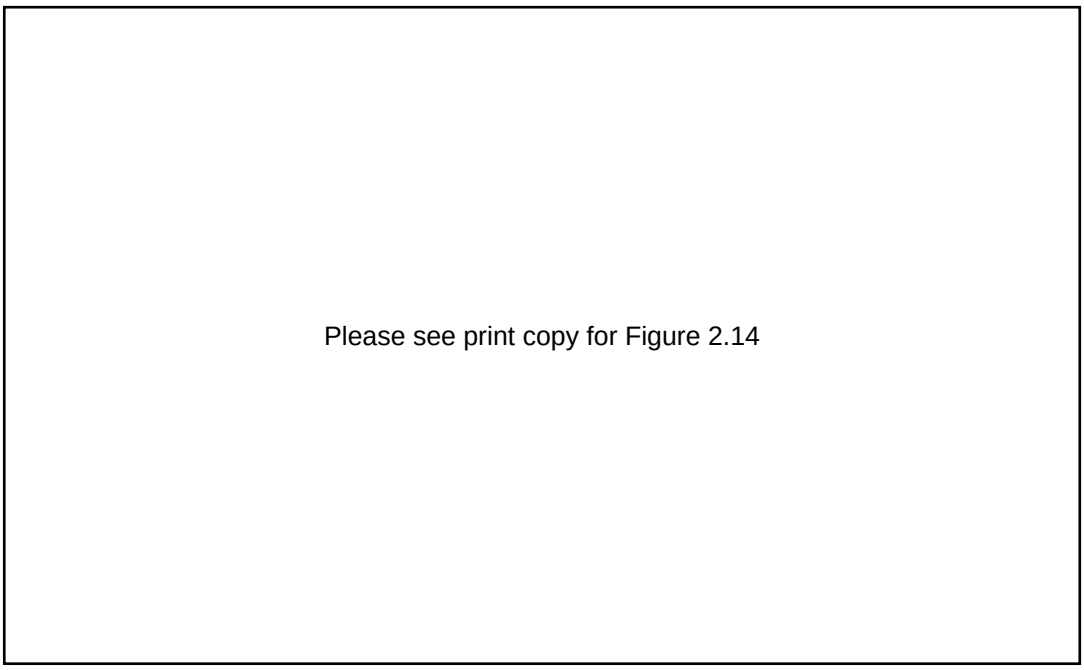

Figure 2.14. The Ohio Supercomputer Centers' surgical simulator for temporal bone dissection [181].

Real-time visualisation and haptic rendering of volume cutting in petrous bone surgery has been accomplished at the University of Hamburg, Germany [74, 111, 119, 120, 128, 129] (Figure 2.15). The authors [74, 111, 119, 120, 128, 129] combine a multi-volume representation of the image data and interactive cutting tools with a sub-voxel resolution surface construction technique. Interaction forces are calculated at this resolution using a multi-point collision detection algorithm and transmitted to a 3DOF haptic device, achieving an update rate of 6000Hz [111, 119, 120, 129]. Haptic rendering is based on the same ray-casting algorithm that the authors use for visualization of movement and surface recognition, based on voxel densities and the partial volume effect [74, 111, 119, 120]. This is combined with adaptive sampling for

the removal of artifacts during volume visualisation [74]. Results reveal superior quality graphics and detailed haptic resolution [74, 111, 119, 129]. A sphere-shaped tool with sample points equally distributed over its surface represents the otologic drill [111, 119, 129]. During temporal bone removal, forces are calculated relative to points that come into contact with an object [111, 129]. The mastoid bone is assumed to be a rigid structure and irrigation is not included, nor are its effects on temporal bone properties examined in the simulation [111]. Soft-tissue deformation is also neglected [74, 129]. Stereoscopic viewing is included to enhance the realism of the simulation [74, 129].

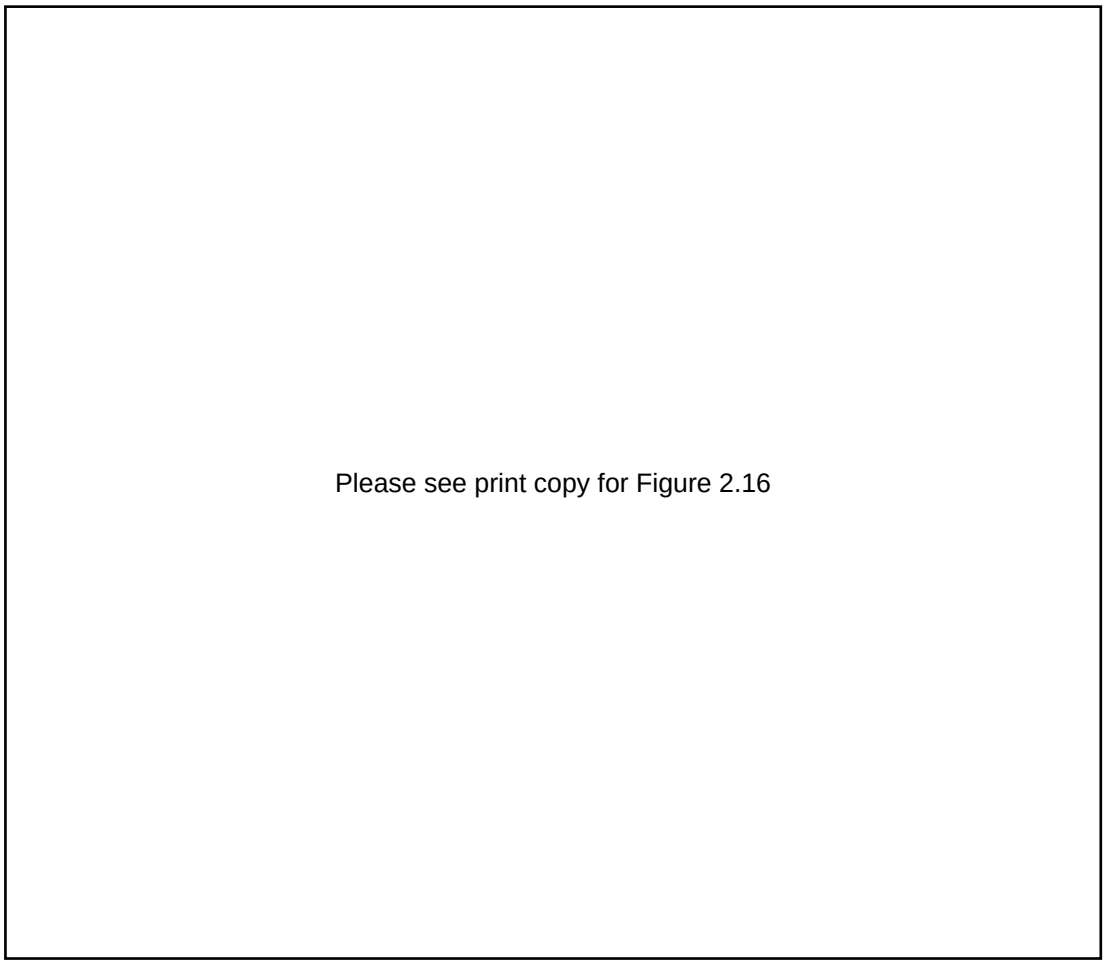

Figure 2.15. University of Hamburg's temporal bone dissection simulation [74]. Bone drilling reveals anatomical structures of the middle and inner ear.

A volumetric representation of the middle ear has been derived from patient-specific CT and MRI data for the purpose of modelling burr-bone interactions during a mastoidectomy [116, 121-125, 130, 131]. The simulation is shown in Figure 2.16. Burr-bone interactions, accumulation of bone dust and fluid movement due to irrigation are synthesized for training purposes [122, 131]. The authors [123] model tool/object interactions using a volume sampling algorithm, which is based on texture mapping and back-to-front composition of the image slices. Force calculation is a function of burr position and voxels are identified as being bone, empty or bone paste [116, 123]. Force computation is based on an erosion model, where voxel densities are decreased when the bone voxels are in contact with the burr and forces are delivered to the user through two haptic devices, at a rate of 1kHz [121, 123, 124]. Agus et al. [123] describe the burring process as comprised of two stages: first, bone deformation and elastic forces are calculated, based on burr and bone positioning, and second, the energy difference between the work of the drill and that needed to cut the bone is determined, to find the burr cutting rate [124]. Both elastic and frictional force contributions are computed in the simulation, the summation of which gives the total output force [122-124]. A hybrid particles-volumetric model is adopted to simulate water flow during irrigation as well as formation of bone paste upon bone and water particle collision [124]. The authors [124, 125] consider that further verification of the forces involved during drilling is required, based on experimental data.

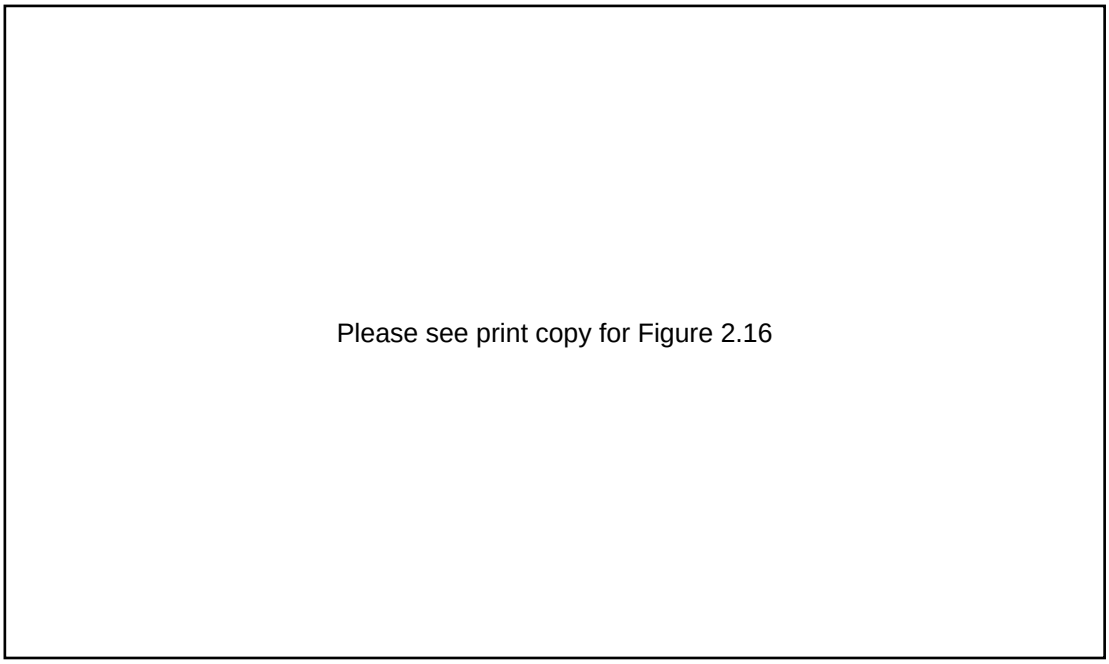

Figure 2.16. Simulator of temporal bone drilling, with irrigation and suction, developed at the Centre for Advanced Studies, Research and Development, Sardinia, Italy [125].

The CSIRO, Australia, have also produced a temporal bone drilling simulator, for networked instruction [126]. Volumetric models of the bone are produced for dissection by the user, with polygonal surface models of some of the surrounding anatomy [126]. A similar approach has been implemented by Kühnapfel et al. [127].Two haptic devices are used for interaction with the model, via a range of tools including drill bit, irrigator, sucker, guiding arm and marker [126]. The authors concede that the physical (haptic) modelling requires further attention to increase accuracy in this area [126].

2.6.2 Soft Tissue Manipulation

Soft tissue manipulations, including tissue deformation [100, 102, 132, 137, 139, 140, 149, 150, 156, 158, 164-169, 171-173, 178, 180, 182-185], cutting [150, 159, 160, 164,

174, 177, 179, 182, 185-188] and/or device insertions [132-145, 147, 148, 158, 170], have been visually and haptically rendered in the virtual operating suite. Arbitrary incisions and surface deformations of this type are not easily modelled, particularly when haptics and volumetric datasets are involved. Real-time constraints imposed on the system often lead to model simplification, such as the assumption of rigid body dynamics. The most common approaches for modelling the physical behaviour of anatomical objects include FEM [100, 102, 110, 132, 138, 147, 148, 150, 168-170, 172, 178, 183, 184, 186], mass-spring models (MSM) [127, 139, 149, 156, 157, 159, 164, 165, 174, 175, 179, 180, 187, 188] and meshless schemes [100, 173, 177].

A FEM approach is compared with a meshless, multi-point based technique of the Method of Finite Spheres (MFS), for simulation of stomach palpation (feeling the organ with the hands) [100]. Results show that the MFS significantly reduces computation time [100]. Physical properties, including young's modulus of the stomach, are added to the geometric, polygonal model [100]. Shape functions are used to determine surface deformations and stiffness matrices are applied for the computation of reaction forces delivered from the simulation [100]. A hybrid approach is used to model soft tissue deformation [167], where a FEM of the liver is manipulated with a probe tip which is surrounded by a set of nodal points that deform the mesh. Organ palpation and subsequent deformation is modelled using the Atomic Unit (AU) method, where force is delivered throughout the organ which is modelled as a series of finite volumes [165]. A surface-based mass-spring-damper model is used to calculate the input force and the output force is computed which is proportional to the probe distance traversed within the constructed force-field [165]. The spring constant of the model is updated in real-time, as the force-distance calculations vary [165]. Collision detection is implemented using an octree-based scheme; where leaf nodes represent either empty or

full voxels [165]. Visual and haptic rendering of palpation is also simulated for prostate cancer diagnosis [146]. A polygonal model of the prostate was generated for deformation, where the degree of nonlinear deformation is based on the location or penetration of the tool, relative to the surrounding mesh [146]. The physical parameter stiffness is defined by the subjective evaluation of a urologist, based on a priori knowledge of tumour malignancy [146]. Polygonal interactions form the basis for FE collision detection and deformation response for soft tissue rectal palpation [166]. Deformation and force computations are based on tissue stiffness and degree of displacement [166]. A FEM approach is also used for the simulation of breast palpation [168, 169].

Deformation of the liver is simulated using a hierarchical approach to the FEM, wherein only the portion of the soft tissue that deforms is represented by a FEM (termed the child mesh) and the remaining volume (parent mesh) is fixed [178]. This method reduces the computational expense associated with the FEM, whilst maintaining a high degree of accuracy and efficiency [178] which normally would cause computational burden [110] and inhibit real-time response [171]. Pre-computation of elementary deformations and a speed-up algorithm are combined with a FEM to enable real-time liver manipulation [102, 172]. Collision detection for mesh deformation is achieved at 300Hz and a Laparoscopic Impulse Engine is used for force-feedback [102, 172]. Tissue cutting is not realised due to the required pre-computation phase [102, 172]. Manipulation of the liver, such as clamping it with tweezers, is simulated using a meshless method that defines the organ as a myriad of spheres [173]. Haptic sensation is delivered to the fingers using two custom force feedback devices, specifically designed for interaction with the liver model [173]. A rendering of tissue deformation of the liver is shown in Figure 2.17. Deformation during human thigh examination is visually and

haptically rendered [149]. A meshed-based spring-damper model is used to physically represent the model and a simplified buffer model is derived for real-time force response [149].

A custom 4DOF haptic device (including 3DOF force output: pitch, yaw and insertion, 1DOF rotation for viewing) is developed for force delivery during simulation of knee arthroscopy (internal examination of the knee joint) [150, 151]. Real-time volume deformation and cutting is realised using hybrid FEM, where interaction is monitored between the cutting tool and local tetrahedra with which the intersection is taking place, and temporary subdivision of a tetrahedron takes place until it is completely cut and then discarded [150, 151].

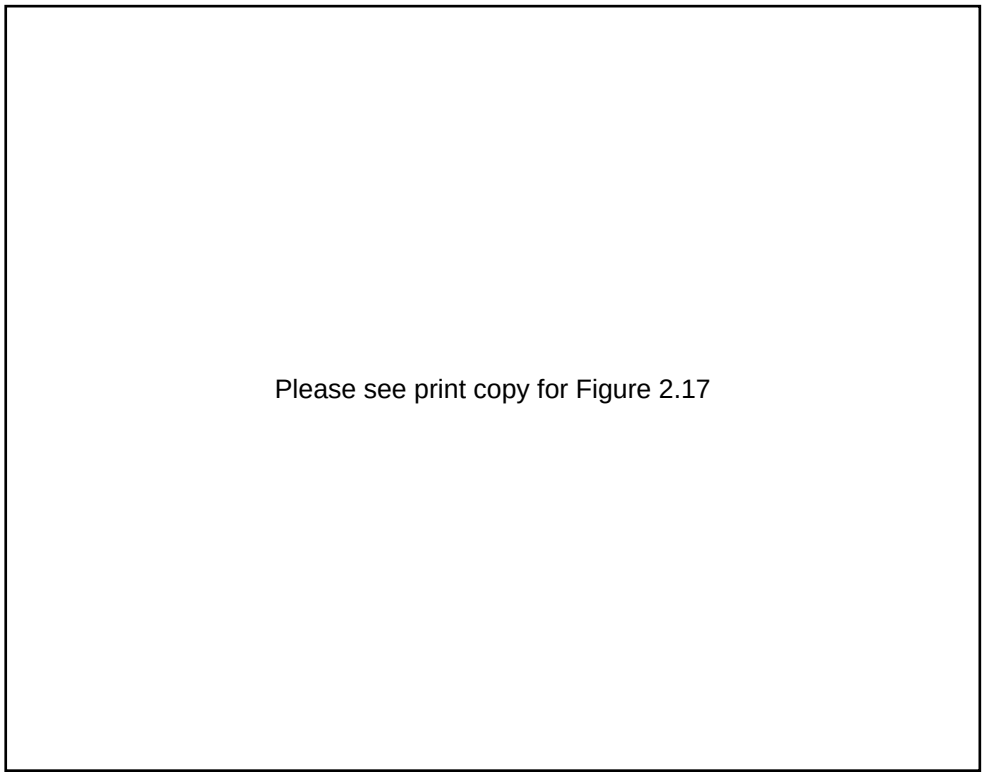

Figure 2.17. Deformation of the liver [164].

MFS is used in conjunction with a local polygon subdivision algorithm and node snapping method for the simulation of soft tissue (stomach) cutting [177]. This technique requires only local interaction between the tool and its environment, rendering it a less computationally expensive exercise than a FEM for representation of the entire structure [177]. Manipulation of the skin, including tissue cutting and clamping, is implemented using a mass-spring model, for the purpose of hernia treatment [174]. Soft tissue deformations during tearing and pulling for cataract surgery are recreated using mass-springs-damper models within a meshed surface topology [179]. Mass-springs-damper lattices are also adopted for generic tissue cutting with scissor apparatus [175]. A hybrid approach, combining a FEM with tensor-mass elements, is derived for soft tissue cutting and deformation [182], since pre-computation associated with the FEM in the work did not allow for arbitrary incisions. An elastic model of the liver is produced for removal of the liver (hepatectomy) with two force-feedback devices that represent the tools used for laparoscopic surgery [182]. Bro-Neilson et al. [164] use mass-spring models to physically represent the stomach during deformation (Figure 2.17) and cutting (Figure 2.18). In this system, blood flow is also modelled. A FEM is disregarded for this purpose due to the complexity associated with remeshing during the cutting operation [164]. A triangular topology is used with mass-springs to enable tissue dissection, for a range of arbitrary incisions [164]. The authors use inbuilt collision detection and response algorithms in GHOST SDK as a basis for more complex force modelling during model manipulation [164].

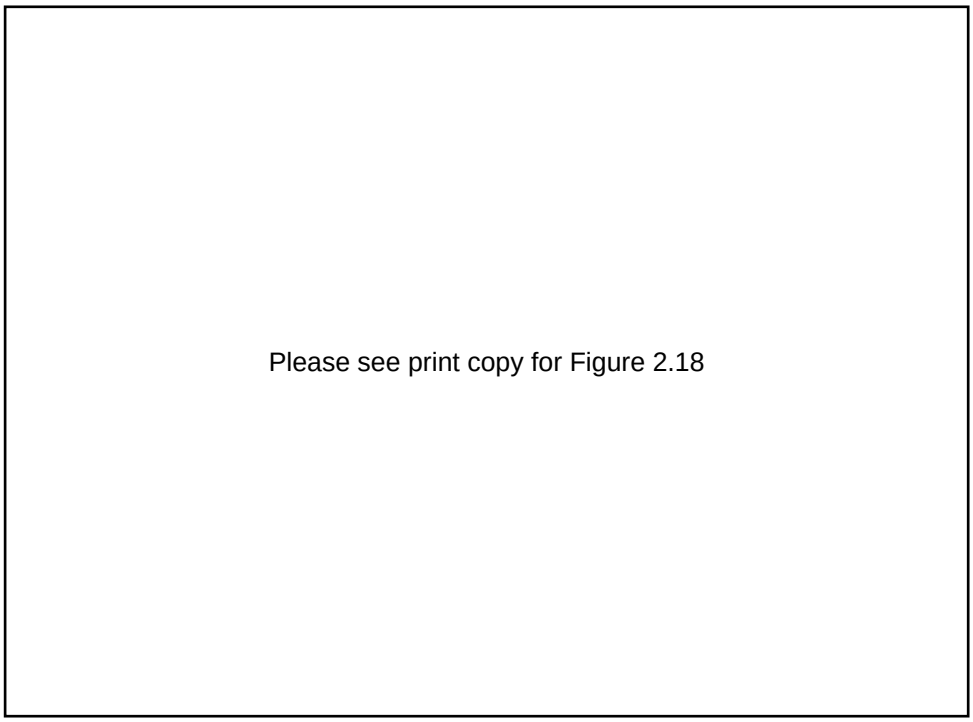

Figure 2.18. Soft-tissue cutting to reveal the stomach [164].

2.6.3 Device Insertions

Devices such as needles, catheters, endoscopes, laparoscopes and prosthetics may be inserted into the human body for the purpose of material removal, repair or examination. Soft tissue needle and catheter insertions have been simulated, for the purpose of medical training [132-134, 136-139, 142-144, 148] and force analysis [132, 138, 140]. DiMaio and Salcudean [132, 138, 140] use FEM to represent soft tissue as a two-dimensional mesh, for the simulation of needle insertion. In this application, a higher concentration of mesh elements exist around the needle location and as the needle is inserted into the model, the node positions change to model the tissue deformation [132]. Adaptive meshing is introduced, where node placement changes according to the needle trajectory for visual rendering and node snapping to the needle location is used for haptic rendering [140]. Insertion force experiments are performed to validate the

simulation [132, 138, 140], where contact forces along the needle are integrated to give a total force output [140]. Needle insertion is simulated for training in spine biopsies for tumour evaluation [133, 134]. Tissues are modelled as springs, whose stiffness changes with insertion depth and biological properties are derived from experimental data [134]. Force calculation is dependent on needle position, tissue stiffness, needle velocity and insertion depth [134]. Forces are delivered back to the user through the Phantom Desktop at a rate of 1kHz [134].

Kyung et al. [133] first assign stiffness values for force calculation during needle insertion. The authors [133] then use a more progressive approach, implementing a viscoelastic model, where puncture forces are derived from MRI data and relate to changing values for resistive force during needle insertion. Implantation of radioactive seeds for the treatment of prostate cancer is performed using via needle insertion, in a process called *brachytherapy* [137]. Visual rendering of the soft tissue deformation during this process is simulated using a modified 3D ChainMail algorithm [137], with a direct needle insertion interaction technique using a non-linear Gaussian transmission function to model needle/tissue dynamics [137]. Total force delivery is an accumulation of the force due to gravity, a force due to linear constraint (including contributions from friction and damping) and a surface penetration force [137]. Force computation is based on needle position and trajectory. A modified GJK collision detection algorithm (created by Glibert, Johnson and Keerthi [139]) is implemented to determine contact between the needle and polygon surface topology [139]. Upon collision, mass-springs produce output forces [139]. The marching cubes algorithm is applied for three-dimensional surface reconstructions of the liver, kidney and neck, for the purpose of needle insertion [142]. Surfaces are converted to VRML representations and in the simulation, the original X-ray and CT image data are superimposed on the structure

[142]. Forces are calculated as the needle penetrates the skin layer: these are based on whether the needle has penetrated the voxels that correspond to the soft tissue, from the original image intensity values [142].

Zorcolo et al. [143, 144] combine back-to-front volume rendering for patient-specific anatomical visualisation with a physical model of tissue deformation for force computation during a catheter insertion [135]. The latter relies on user-selectable tissue parameters and needle shape attributes. A spring-damper model was originally trialled for tissue representation, but the model did not respond realistically (produced large deformations) to high input forces [143, 144]. An incremental viscoelastic model was instead implemented [135, 143, 144]. Surface breaking forces are awarded to each tissue layer and reaction forces are computed via numerical integration during needle intrusion [135, 144]. A catheter insertion was subjectively evaluated by an experienced surgeon who deemed the visual and haptic sensations realistic [143, 144]. The system, including a screen shot of the catheter insertion, is shown in Figure 2.19.

Catheter insertion is simulated for a coronary stent implantation [147]. A cardiologist performs the operation on a blocked coronary artery: the artery is opened and the stent is implanted so that it remains in a fixed position to keep the artery open following surgery [147]. Visual and haptic feedback are provided to the user during catheter insertion into the artery [147]. Real-time collision detection and response are performed between the artery walls, modelled as a triangular polygonal mesh, and the tool tip, based on its relative position. Forces are delivered back to the user through a custom HERMES haptic interface [147]. Soft tissue deformations of the artery walls are physically represented using FEM and linear elasticity theory [147]. Rates of deformation are calculated by linearly combining pre-computed deformations with real-time responses [147].

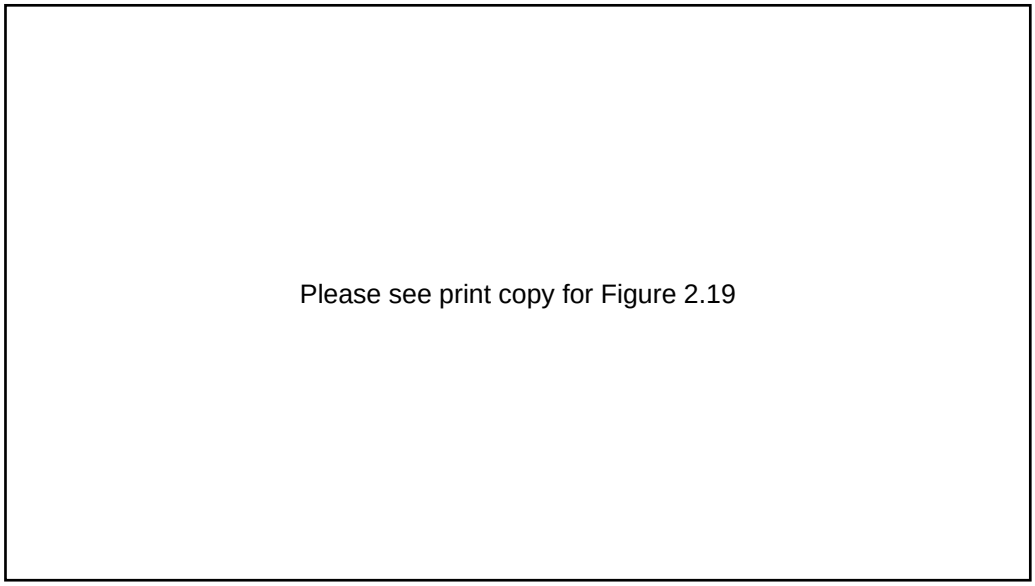

Figure 2.19. A virtual catheter insertion, with stereo viewing and force feedback [144].

An epidural block, which involves placing a catheter into the spine via needle insertion, is visually and haptically rendered [136]. A polygonal surface representation of the anatomy is derived from MRI and graphical rendering of the insertion of the needle is performed using a derivative of the marching cubes algorithm [136]. Force computation is based on a look-up table, with values related to needle insertion depth, yet at the time of publication, work was being done on spring-based force modelling [136]. Forces are delivered through a 1DOF haptic device [136]. Lower back puncturing is performed using a needle that is attached to a force feedback device which is inserted into a mannequin [145]. Tissue variation corresponds to change in output forces, as the needle is inserted through the different tissue layers [145]. Suturing, which involves needle insertion into the skin, is modelled [170] with two-handed user interaction: one hand holds the virtual forceps and the other directs needle movement. A FEM is used to

represent skin and soft tissue as meshed elements [170], whose dynamic behaviour is governed by a set of linear equations with a pre-defined stiffness matrix.

Laparoscopy is an exploratory medical procedure where a tube with a tiny video camera at its end is inserted into the abdomen. Zhang et al. [159] simulate tissue dissection which is used in laparoscopic surgery and model the L-hook laparoscopic cautery tool. A surface mass-spring model is used in conjunction with a mesh-based approach for tissue cutting [188] and an Immersion Laparoscopic Impulse Engine (Immersion Corporation) provides force feedback to the user, with haptic sensation delivered to both hands [159]. Laparoscopic hernia surgery requires the act of stapling [157]. This type of surgery is recreated so that the surgeon can practice stapling two individual deformable meshes together in a virtual environment [157]. The meshed models are represented as mass-springs in a lattice work, where the spring forces and motion are governed by the laws of physics [157]. Change in lattice node position during deformation was done via Newton-Euler integration, while fixing the mass points was achieved using velocity compensation methods [157]. Such mass-spring models offer simplicity and processing speed, yet may be limited by stiffness and stability issues [157], as well as model inaccuracy [178].

Tissue cutting and bleeding associated with laparoscopic surgery is recreated [160] with simple haptic algorithms based on spring-damper behaviour. Laparoscopic surgery for bile duct exploration during a cholecystectomy (gall bladder removal) is simulated, with laparoscopic forceps used for catheter insertion [158, 161]. Visual and haptic rendering of the procedure is done by combining a FEM with a particle model, wherein springs and dampers are used to represent the catheter tool [158]. Dedicated laparoscopic hardware is fitted to the commercially available Phantom haptic devices for realistic force reflection and model manipulation [158].

Endoscopy is similar to laparoscopy and has been simulated with haptic feedback for the purpose of surgical training [152-156]. Custom haptic interfaces have been developed for the specific application of endoscopic training [152, 153, 176]. The system produced by Korner et al. incorporates the tube and tip of the endoscope, which generates elastic and frictional forces during interaction with the virtual anatomy, such as the colon [152, 153]. For interactive haptic modelling, a FEM is combined with a beam bending theory to physically model the bending of the colon during a virtual colonoscopy [189]. The colonoscope is represented by spring-dampers distributed along its length [189]. A locally developed, 7DOF haptic interface, the *HapticIO*, is developed for the application of endoscopy [127]. The system has two force-feedback devices: one for the endoscope tool (with camera view) and the other for auxiliary instruments such as scissors or a catheter [127]. This device is combined with previously developed software, *KisMo* [190-192], for haptic and visual rendering of a virtual endoscopy [163]. *KisMo* realises soft tissue deformation via mass-spring modelling, which is based on mass, stiffness and damping properties of the material [191]. Functionality includes grasping, cutting, clipping, coagulation (which includes the production of smoke and steam), irrigation, suction and bleeding [191, 192]. Hierarchical collision detection methods are applied, with the simplification of object geometry to produce real-time responses [191]. Applications for this system have included a cholecystectomy (Figure 2.20), ventriculostomy (forming an opening in the ventricle), as well as temporal bone drilling in which system modifications were required [127]. In contrast, rudimentary haptic feedback is provided by a joy-stick in conjunction with high fidelity visual replication of endoscopy by means of ray-casting [155].

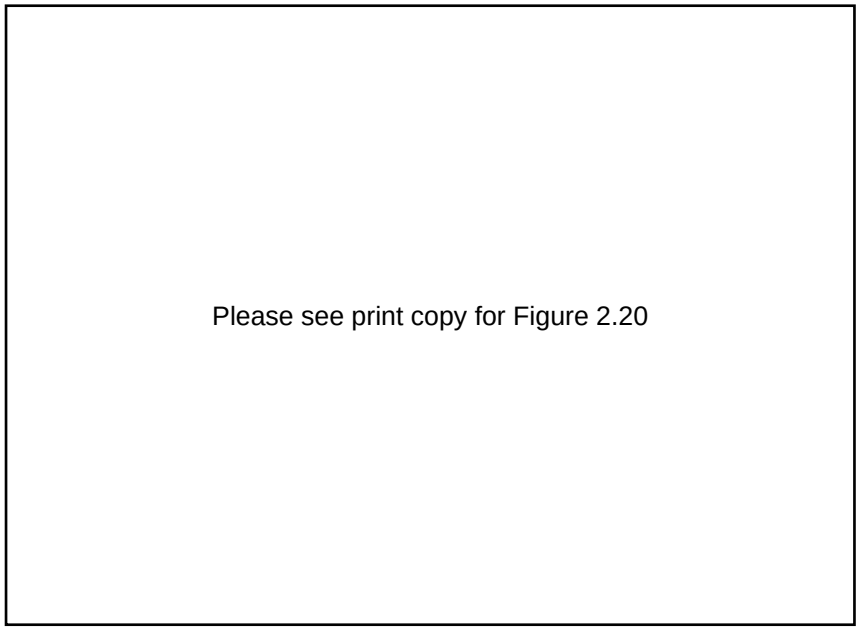

Figure 2.20. A virtual cholecystectomy [127].

2.7 Significance of the Work

In the previous section, a critical review of related work in the field of haptic-rendered surgical simulation was presented. The analysis reveals that whilst there is growing interest in this area of research, there is currently no surgical simulator that offers visual and force rendering during a virtual CI insertion. A simulator that models in real time the physical, dynamic behaviour of a specific design of electrode array as it is inserted into a replica of the human ST has not previously been developed. Real-time force rendering of this procedure is unprecedented. Reasons for this may include:

- Previous limitations in computer technology, including processing speeds, hardware functionality and image resolution.

- The small size and complex shape of the cochlea which makes it difficult to model. Further, the difficulties faced in accurately segmenting the ST from the cochlea as a separate cavity, for CI insertion.
- Precisely calculating the collision detection and response of the complex interaction between the ST walls and the flexible silicone housing of the array.
- Real-time constraints on the haptic update rate, requiring a high speed. This is difficult to maintain with accuracy for complex modelling scenarios such as cochlear implantation.

CI insertion is unlike previous simulations such as temporal bone removal, soft tissue incisions or deformations. Although three-dimensional reconstructions of human cochleae have been produced for the purpose of measurement, visualization and/or animation (including endoscopies and implant insertion), there remains no application in which force feedback is relayed to the user during a real-time CI insertion. A research deficiency has therefore been identified, which is the basis of motivation for the work in this thesis. Since the first cochlear implantation in 1978, there have been rapid advances in computer technology. With the current software and hardware capabilities it is reasonable that this type of surgery can now be modelled to a high degree of accuracy. By utilising available resources, the work focuses on modelling CI insertion and particularly, accurately rendering the force feedback during this procedure, the latter of which forms the basis of this thesis.

The most important and unique contributions of the work include:

- Production of an anatomically accurate, three-dimensional model of the human ST. The model is parametric and reproducible so that patient-specific models may be derived using the same technique.
- Accurate, real-time force delivery during a virtual CI insertion.
- Validation of the simulator output by experimental force measurement.
- A training program for the surgeon to practise CI insertion on a low-cost machine and supporting software, within a risk-free virtual environment.

In the next chapter, insertion studies are performed to evaluate force delivery and electrode trajectories for different implant designs and insertion techniques. This is done in order to accurately design, construct and finally, validate the surgical simulator for cochlear implantation.

Chapter 3

INSERTION FORCE ANALYSIS

3.1 Insertion Force Studies

In order to design, construct and validate a haptic-rendered surgical simulator for CI insertion, an analysis of insertion forces, administration technique, electrode design and its location during device implantation is required. Without an accurate description of the insertion process, the procedure cannot be precisely replicated. In this chapter, force administration, electrode trajectories and surgical technique during cochlear implantation for the Nucleus® 24 Contour™ and Nucleus® Contour Advance™ electrodes are examined. Experimental results that have been produced from this work, including insertion force profiles and coefficient of friction measurement between the electrode/ST interface, will be used to design and validate the haptic-rendered surgical simulator for CI insertion. The related literature does not report of any quantitative analysis on force administration during implantation of the Contour or Contour Practice electrodes using the Standard Insertion Technique (SIT) or partial stylet withdrawal method. Furthermore, there is no existing work on force delivery during insertion of the Contour Advance electrode, which is uniquely implanted using the Advance Off- Stylet (AOS) technique. In this study, the design of electrode and advancement methods are considered for the explicit purpose of modelling a typical insertion in a virtual environment.

During the literature review, the existing work on the measurement of forces

involved in CI insertion will be examined. Existing differences in the design of electrodes deployed in this work will be identified. Experimental methods and results are presented for the SIT, partial withdrawal and AOS insertions. Force delivery and electrode trajectories for each process are assessed and the key factors contributing to force administration are considered. Furthermore, the optimal technique for minimising force output during implantation is discussed. Finally, the electrode design and associated technique to be synthesised in the work will be selected.

3.2 Background

Insertion studies have been performed to quantify force delivery [35, 36] and evaluate electrode trajectories [26, 35, 36, 46] during simulated cochlear implantation. These were carried out to assess force administration for different electrode designs, including the Clarion (Advanced Bionics Corporation) [35, 36], the Combi C40+ electrode (Med-El) [46], new prototypes such as the Flex EAS [46] and reduced-element electrodes [35, 36]. Researchers have identified key elements that affect force output during electrode insertion, including implant design and insertion behaviour [35, 36], restoration forces [35], carrier stiffness [26, 35, 42], contact pressure at the tip [26], as well as the coefficient of friction between a lubricated model and silicone tip [36]. Synthetic models of the ST have been used for the purpose of insertion force measurement during cochlear implantation [35, 36, 46]. Existing studies have shown the importance of quantifying force components and the assessment of electrode trajectories for specific electrode designs, for the minimisation of trauma [11, 25, 30, 31, 35-39, 42-44], optimal electrode placement [3, 14, 25, 30-35] and to avoid electrode damage [37, 39, 42].

Insertion studies were carried out to assess force administration for different electrode designs and insertion methods. Rebscher et al. [35, 36] examined insertion

forces and electrode positioning during advancement of a reduced-element electrode (silicone housing without individual electrodes or wires) into a three-dimensional acrylic model of the ST. The work was undertaken to assess implant design and insertion behaviour, including electrode positioning, for overall improvement of the electro-neural interfacing and to minimise trauma associated with the procedure. Insertion of the Clarion design (Advanced Bionics Corporation) was also done for comparison of results. The synthetic model used by Rebscher et al. [35, 36] was derived from cadaver specimens. An actuator was employed to advance an insertion tool which held the electrode, into the fluid-filled model of the ST and a 50g force sensor was mounted below the model for data monitoring. Insertion force profiles were provided by Rebscher et al. [35] where force was plotted against distance traversed, along with images taken of the electrode inside the model, for several stages of the process.

Restoration force and carrier stiffness were also determined by the authors [35]. A mechanical strain gauge was used to measure the deflection force required to straighten the carrier from its pre-curled state, about its horizontal axis. Stiffness properties were calculated using the strain gauge to measure vertical deflection of the electrode array carrier at intervals along its length. A load cell and mechanical strain gauge were employed to determine the coefficient of friction for a variety of test lubricants between a silicone tip and epoxy base [36]. Scanning Force Microscopy (SFM) is another method used for coefficient of friction measurements at the micro-level [193]. Combining the results from the previous work [35, 36] provided an overview of the types and magnitudes of forces involved in the interaction between the silicone carrier and ST model during insertion for two types of electrodes: a custom prototype and the Clarion array.

Insertion forces were investigated by Adunka et al. [46] for a new carrier prototype, the Flex EAS, in comparison with the Combi C40+ electrode (Med-El), into a synthetic model derived from radiographic images of a human ST. As the electrode was inserted into the acrylic model (lubricated with medical grade silicone oil [46]) using equipment from Lloyd Instrument Ltd, insertion force was measured with a scale (Precisa Instrument) attached to the electrode. From the data collected by the load cell, force profiles were generated for each type of implant. In a separate analysis, the authors examined electrode positions, insertion depths and degree of trauma by histological examination of human temporal bone cadavers following device insertion [46].

Chen et al. [26] performed insertion studies using FEM to examine the effect of a virtual change in stiffness gradient of the Nucleus Straight electrode on electrode trajectory and contact pressure at the tip of the carrier. Stiffness profiles for the Nucleus Straight and Contour electrodes were experimentally determined using flexural bending tests, for comparison of stiffness properties between the two designs [42]. The authors [26, 42] suggested that greater carrier stiffness, specifically at the tip, may increase contact pressure in this area and induce trauma.

Existing literature that provide an assessment of force delivery during cochlear implantation highlight the importance of this research. In this work, we examine and compare force delivery, electrode positions and surgical techniques for the Contour and Contour Advance electrodes not previously considered. Experimental results and analyses produced in this section will be then used for development and validation of the simulation model. Coefficient of friction measurements will be utilised in physical modelling of the ST/electrode carrier interface of the simulation.

3.3 Insertion Techniques: Contour™ Electrodes

Specialist technique will vary depending on the type of implant that is chosen for insertion. In this analysis, trajectories and force delivery for implantation of the Nucleus® 24 Contour™ and Nucleus® 24 Contour Advance™ arrays with the insertion methods that are currently used by specialists are examined. The SIT or partial withdrawal method is performed using the Nucleus® 24 Contour™ and an AOS insertion is uniquely associated with the Nucleus® 24 Contour Advance™. The electrodes are approximately 22mm in length (from tip to first rib), tapering from a tip diameter of about 0.5mm (0.4mm Advance) to 0.8mm. The silicone-coated electrode is kept straight by a thin platinum wire (stylet). For both designs, stylet withdrawal will result in the arrays returning to their pre-curled state.

The Nucleus® 24 Contour™ was introduced by Cochlear™ in 1999 (Chapter 1, Figure 1.3). For this type of electrode, the surgeon uses either the SIT or partial withdrawal method. For the SIT, the electrode is advanced into the cochlea using tweezers, with the stylet in place. The electrode is fully inserted to the second rib of the carrier (Figure 3.1, point 4) and the stylet is then withdrawn (Figure 3.1, point 5). The surgeon may combine the SIT with a partial removal of the stylet around the Basal turn area, which would correspond with point 2 in Figure 3.1. Using this approach, the electrode is inserted until the surgeon feels the point of first resistance and the stylet is partially removed (approximately 1mm to 2mm [31]). Continuation of advancement after this point without partial removal of the stylet may inflict trauma in this area [37, 39]. The electrode is then fully inserted (Figure 3.1, point 4) and the stylet withdrawn (Figure 3.1, point 5). The Contour is pre-curled prior to insertion and as the stylet is withdrawn, the array recoils. If positioned correctly, it is oriented towards and lies along the inner wall. Using this technique, the electrode does not touch the outer wall and is

positioned closer to the modiolus, exerting restoration forces against this axis (Figure 3.1, point 5).

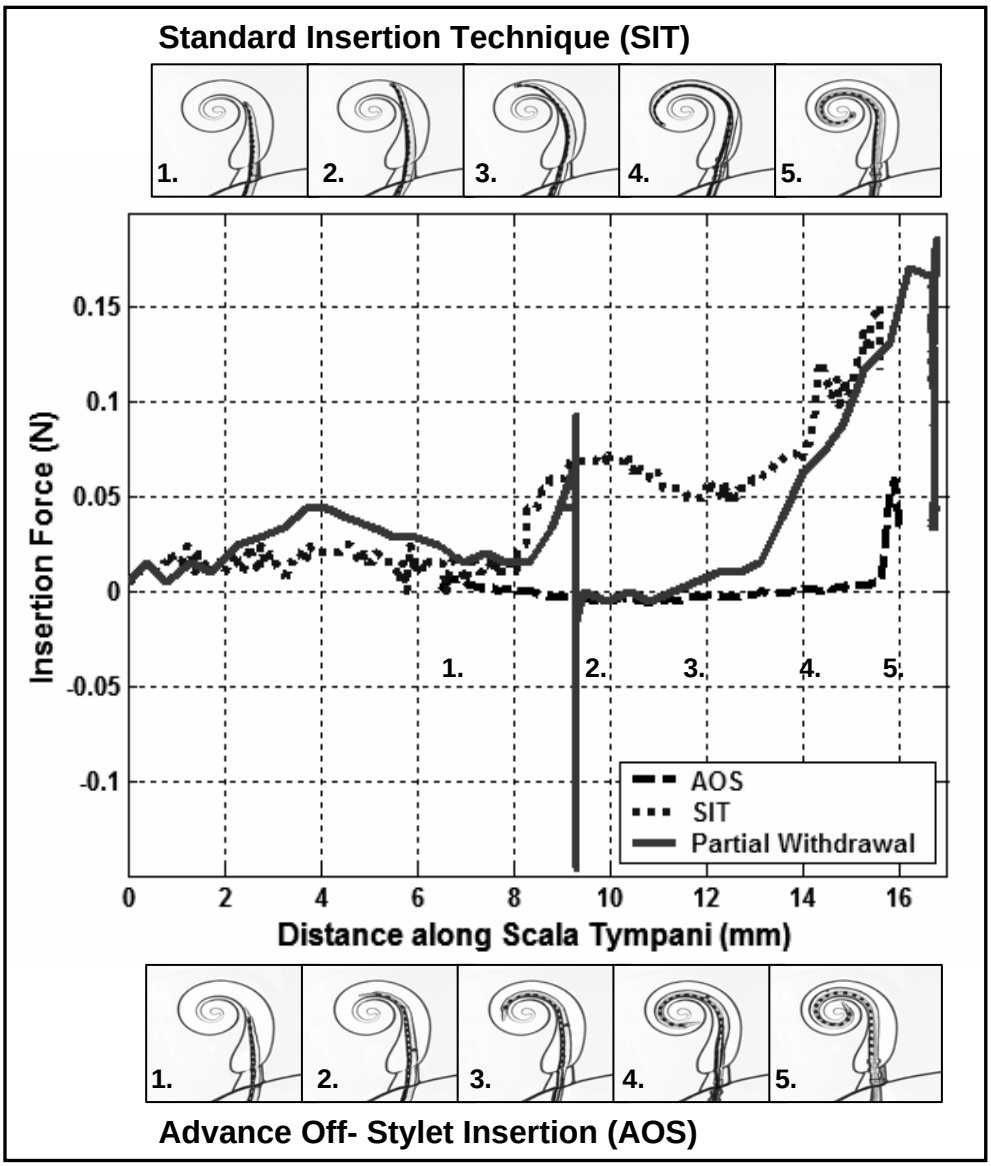

Figure 3.1. Electrode trajectories (SIT and AOS insertions) and force profiles for a typical SIT, partial stylet withdrawal and an AOS insertion. The different stages of insertion are numbered 1. to 5., for the Contour and Contour Advance electrodes, from left to right as the array is inserted further into the cochlea. Spikes in the output force for the partial withdrawal method at 9.5mm and 16mm marks are due to the stylet withdrawals.

The Nucleus® 24 Contour Advance™ electrode was designed with a soft-tip, to enhance flexibility in this area (Figure 3.2). Developed in 2004, this electrode is similar in size and shape to the Contour array. The main difference is the soft-tip. It is a silicone structure of reduced hardness and different geometry compared to the Contour tip. As part of its design, the Contour Advance has a white line (marker) on the silicone housing, approximately 9mm from its tip, to provide the surgeon with a visual aid during electrode advancement. A distinctive technique is used for insertion of the Contour Advance: the AOS insertion. Using this approach, the electrode is inserted with the stylet in place until the white marker on the carrier is aligned with the cochleostomy site (Figure 3.1, point 1). The surgeon clasps the carrier with tweezers using the dominant hand and holds the stylet loop with tweezers in the contralateral hand. The stylet is held in place, while the electrode is advanced off it and into the ST. As it is inserted, the electrode assumes a pre-curled state and hugs the modiolus, with a final position along the ST inner wall (Figure 3.1, point 5).

In this study, net insertion forces involved during implantation of the Contour and Contour Advance electrodes into a synthetic replica of the ST were analysed. Three techniques were applied: the SIT, partial withdrawal and AOS, using Contour electrodes for the former and Contour Advance arrays for AOS insertions. This information was used to evaluate force delivery and electrode trajectories for each process, to produce and validate the surgical simulator. Analyses were also performed to assess surgical technique and to minimise trauma caused by excessive force application. Studies were performed to identify factors that affect force output during an insertion.

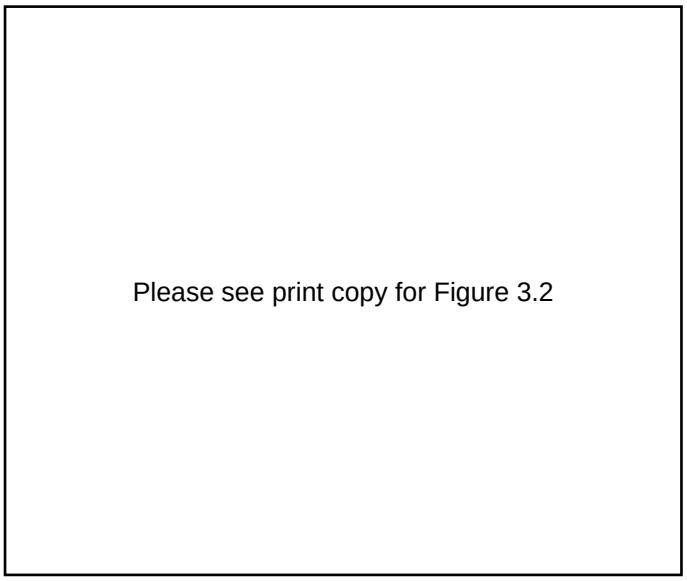

Figure 3.2. Nucleus® 24 Contour Advance™ electrode (Picture provided by Cochlear™). The Contour array is shown in its pre-curled state, following partial stylet withdrawal. Inter-rib spacing of 0.5mm is shown on the diagram to provide an indication of electrode scale.

3.4 Insertion Force Measurement

During an electrode insertion, there are forces acting on the ST walls and electrode which contribute to the total insertion force. These force components include frictional force, F_F, input force from the user, F_{in} (Figure 3.3), relaxation force of the electrode (due to the recoil properties of the pre-curled silicone) and adhesion forces. In this work, net insertion forces were measured (along the longitudinal axis of insertion) and frictional forces quantified for the interfaces used in the experimentation. The former provides an overall measure of output force at different stages of electrode insertion, which will be compared against results produced from a haptic-rendered simulation of the procedure. By analysing the net insertion force profile and electrode trajectories, it is discussed how frictional force and contact pressure (including electrode strength)

contribute to the total force output during cochlear implantation. A value for the coefficient of friction, as determined by experimentation, will be included as a design parameter of the virtual model of the surgical simulator.

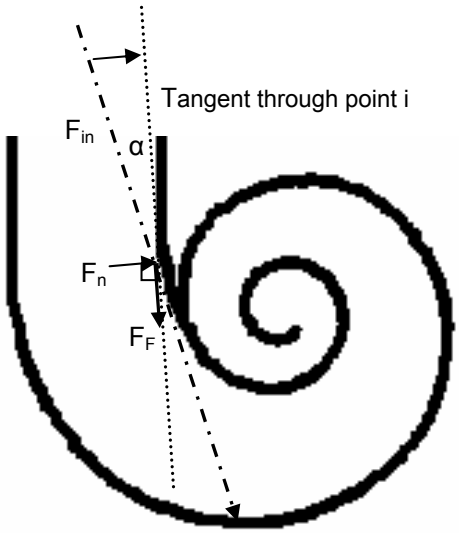

Figure 3.3. A two-dimensional depiction of frictional force at point, i, along the ST inner wall. There is an additive effect of frictional forces during implantation causing an increase in total insertion force.

3.4.1 Model of the Human Scala Tympani

A two-dimensional synthetic model of the human ST, provided by Cochlear^{TM}, was used in this work to carry out insertion studies. The dimensions of the model were taken from published data [11]. Inner and outer wall measurements taken from 11 silastic casts of the ST were plotted in two dimensions as a function of angular displacement about the modiolar axis [11]. The data was used by Cochlear^{TM} to form the cavity of the ST model, which is approximately 9mm (from cochleostomy to Basal turn area) by 6mm (diameter about modiolus), with a depth of 1.5mm. In previous insertion studies,

synthetic replicas have been created from human cadavers [35, 36, 46]. In this work, the model used was machined from Polytetrafluoroethylene (Teflon). Teflon was assumed to have a very low coefficient of friction and with addition of a soap solution, sufficient for modelling the slippery endosteum lining of the ST. Whilst cadaver specimens would have been preferred and may be used in future work for comparison of results, the preparation, storage, handling and acquisition, considering ethics requirements as well as cost, prohibited use of the material. The Teflon model did enable viewing of the electrode trajectory during electrode insertion. At the site of the cochleostomy, the opening was widened (by about 0.1mm) to minimise forces associated with electrode advancement in this area. This replicates the real procedure, where a 1.5mm burr is used to create the cochleostomy and a 1mm burr then used to trim its periphery [25]. This means there will be slight variation in cochleostomy size, as well as position [27], between live cases.

3.4.2 Experimental Procedure

A series of experiments were carried out using a calibrated Instron 5543 force measurement device to advance the electrode into a stationary ST model (Figure 3.4). The insertion studies were performed to evaluate force administration during cochlear implantation of three different electrode designs, that have not been analysed in other work. A load cell was attached to the Instron device to monitor insertion forces associated with cochlear implantation using the Contour Practice, Contour and Contour Advance electrodes (Figure 3.4). Cochlear^{TM} has developed the Contour Practice array for use in surgical training. It is of similar geometry and material composition to the Contour, but has fewer electrode wires and is constructed using a different process to reduce the cost of manufacture. The SIT and partial withdrawal methods were applied

for the Contour Practice and Contour electrodes. AOS insertions were performed exclusively for the Contour Advance.

The sensor was mounted above the upper clamp which held the tweezers and electrode, for the SIT and partial withdrawal methods. This apparatus collectively moves downwards to insert the electrode into the ST model which is held securely in a lower clamp. Using this configuration, the sensor detects forces exerted onto the tweezers that grips the electrode. For AOS insertions, the load cell is mounted below the lower clamp and does not move during electrode advancement (Figure 3.4). In this case, the sensor is mounted below the model to capture forces imparted on the ST itself. This eliminates the effect of forces due to interaction between the stylet and electrode as it is inserted. The length of the electrode is aligned with the initial passage of the ST model, between its inner and outer walls (Figure 3.4, inset **A.**).

Each test was repeated several times, for each insertion technique, the SIT with Practice electrode (5 times) and Contour (11), partial stylet withdrawal for Practice (8) and Contour (26), and AOS (34). Between successive insertions, the stylet was carefully straightened by hand. A specialised tool was used to uncurl the carrier for insertion of the stylet with the aid of tweezers. The electrode was then transferred to a pair of bent-tip tweezers which secured grip at the third marker rib. This module was then attached to the Instron device. A 10N load cell was used to measure insertion forces which were collected at 50ms intervals on the PC by Instron data logging software. Prior to each insertion, the model cavity was lubricated with a soap solution of 10% Bathox and 90% distilled water. This was done to imitate the fluid-filled cavity inside live cochleae and was expected to reduce friction between the carrier and ST walls.

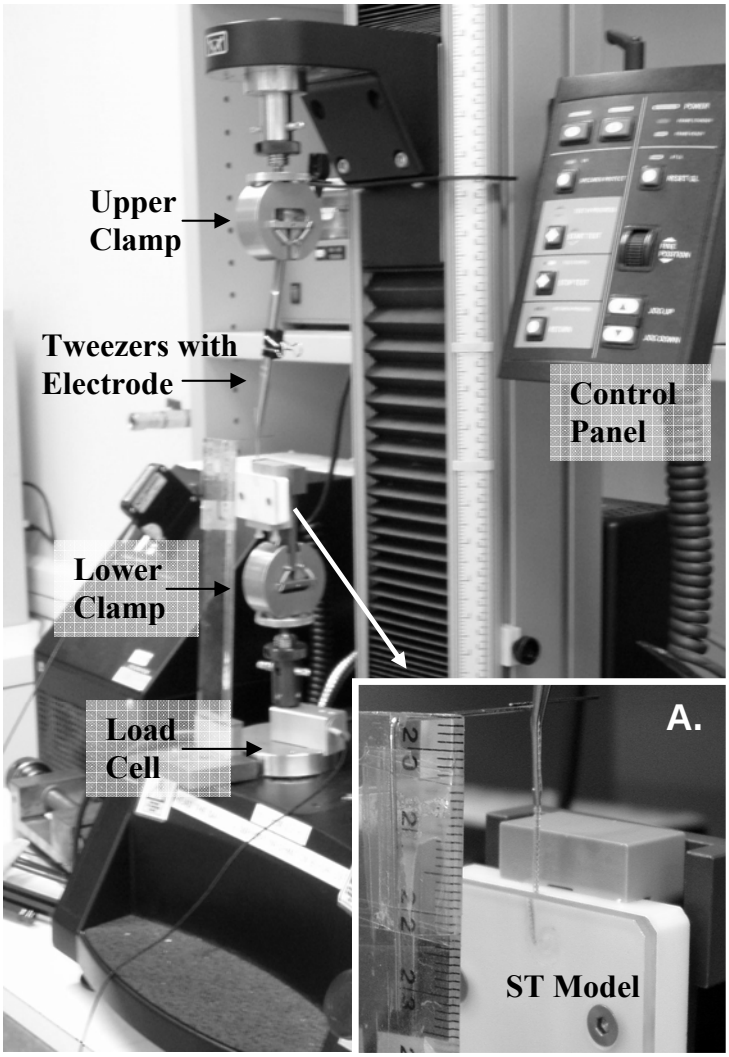

Figure 3.4. Test rig for measurement of insertion force. An Instron 5543 device is used for insertion. The load cell is mounted in a clamp below the model for advancement of the Contour Advance electrode with an AOS insertion, as shown in the photograph, and is mounted above the model for insertion of the Contour electrode using either a SIT or partial stylet withdrawal. A close-up of the electrode insertion is shown at the bottom right-hand corner of the photograph, labelled 'A.', indicated by the white arrow. The magnified view shows the electrode held by tweezers being inserted into the ST model in an AOS insertion.

For the SIT and partial withdrawal approach, the electrode was moved down until the tip was about 1mm to 3mm inside the ST opening. This prevented large spikes in the force profile caused by the electrode tip catching at the entrance. The Contour Advance was inserted to the white marker on its silicone envelope, at a displacement of approximately 6.5mm to 9mm from the tip. It was then fed off the stylet and into the chamber at a constant speed of 120mm/min, until the maximum extension was reached. This was 16mm to 18mm further into the cavity for the Contour and 8mm to 10.5mm for the Contour Advance. At full insertion, the second rib of the array was at the passage opening. Force profiles were generated from the data collected at the PC for each insertion method. An analysis of the results was carried out, to ascertain the effects of electrode design and insertion technique on force output during implantation.

3.5 Coefficient of Friction Measurement

In this work, frictional force was measured to determine the coefficient of friction between the lubricated silicone/Teflon interface. This was done in a separate set of experiments and was deemed as an important parameter to measure, since frictional force contributes to the overall force delivery during cochlear implantation. As the electrode touches the ST inner wall, the outer wall or both (depending on the insertion technique), there will be some degree of frictional force acting between the silicone carrier and ST walls. The values for the coefficient of friction will be used in the simulation. The impact of frictional force on final force delivery during electrode advancement was also assessed.

A Standard CSEM Tribometer (CSM Instruments) was used in a pin-on-disc configuration to measure the coefficient of friction between Teflon and silicone samples, for varying degrees of surface roughness and lubrication (Figure 3.5). Circular

Teflon discs of 25mm diameter and 3mm depth were precisely machined in a lathe and some samples finely polished for a smoother surface. Silicone specimens had either a rectangular surface area or a spherical geometry (of ball configuration, similar to the electrode tip, ~1mm^2 contact area). The silicone was mounted tightly in the tribometer and lowered onto the Teflon disc. Radii about the axis of revolution were changed between trials, measured from the centre of the disc. Speed of rotation was set at 1cms^{-1}, load was varied from 1N to 10N and a sampling rate was set at 10Hz. Experimentation was performed at room temperature. Trials were done with and without lubricant, on rough and smooth disc surfaces. This was done to determine the impact of lubrication and surface roughness on starting and dynamic coefficients of friction for this application. In this context, the starting friction is equal in magnitude and opposite in direction to the force required to set the disc into motion from its state of rest and the dynamic friction is a resistive force that opposes the disc surface whilst it is in motion, which is usually less than the starting friction.

The coefficient of friction, μ, can be calculated by application of (1):

$$\tilde{F}_f = \mu \tilde{F}_n \qquad (1)$$

where F_f is a force due to friction, F_n is a normal force and μ is the coefficient of friction.

A force sensor that was mounted on the head of the tribometer measured frictional force, F_f (Figure 3.5). The normal force, F_n, is the downwards force applied at the head from a known load. Instantaneous values for μ were generated from the measurement of these two values. The Instron software calculated values for μ over time and plotted this result. To get accurate measurements, the device was manually calibrated for each load.

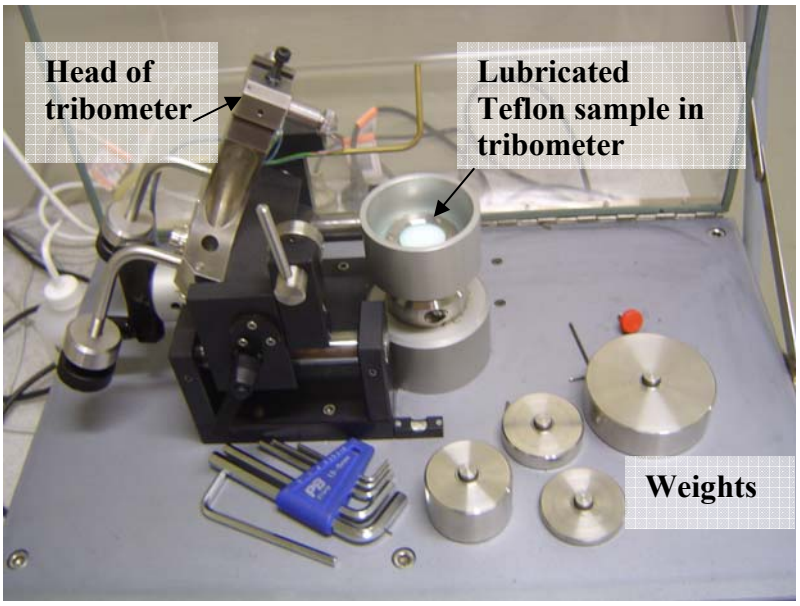

Figure 3.5. A Standard CSEM Tribometer (CSM Instruments), University of New South Wales, Sydney, Australia. With a known weight placed above the silicone sample, the force due to friction at 90º to this normal force can be measured to give a coefficient of friction over time.

In the following section, the results that were produced from the measurement of insertion forces during electrode advancement into the ST model and coefficients of friction (for varying degrees of surface roughness and lubrication) are presented. The significance of these results is then discussed, with an assessment of the parameters that directly affect force output. The scenario that best represents the simulated insertion is investigated.

3.6 Results

Electrode trajectories and force data generated by insertion of the Contour and Contour Advance electrodes are presented in this work, for the SIT, partial stylet withdrawal and AOS methods. Typical force profiles for the SIT, partial stylet withdrawal and AOS methods were shown in Figure 3.1, with electrode trajectories corresponding to specific stages of the insertions. Average output force values and first standard deviations that were calculated from the entire set of insertion forces, for each insertion method, are summarised in Figure 3.6. Starting and dynamic coefficients of friction for the Teflon/silicone interface were quantified and averages are shown in Figure 3.7. These results are intended for comparison of force delivery between the different insertion methods and to identify factors that directly affect force administration, at various stages of electrode advancement.

3.6.1 Insertion Force Output and Electrode Trajectories

(1) Standard Insertion Technique: The SIT was performed for the Contour Practice and Contour arrays. Force profiles that were generated from insertion of the two types of electrodes appear similar in shape and magnitude. As the electrode is inserted into the ST, the total force generally increases (Figure 3.1). There is a small peak around the 4mm mark which is due to the outer edge of the electrode hitting the cochleostomy site. Insertion force tends to drop off after the 5mm mark, as the tip no longer touches the inner wall and the side slides along the outer wall as it progresses towards the Basal turn. A peak in insertion force is observed as the electrode tip touches the lateral wall of the Basal turn, averaging 0.095N and 0.098N for the Contour and Practice electrodes respectively, around the 9mm to 12mm mark (Figure 3.1, point 2.). The net insertion force increases as the electrode array is fully advanced and reaches a peak value where

the first marker rib touches the ST opening just before full insertion, where it drops off slightly (Figure 3.1, point 5). Force profiles exhibit an overall increase to the average peak value of 0.113N (Contour) and 0.09N (Practice) just before the rib touches the cochleostomy site (Figure 3.1, point 4). After this point, forces increase dramatically to 0.194N (Contour) and 0.178N (Practice). The stylet is then withdrawn. During electrode removal from the model, forces increase to 0.037N (Contour) and 0.029N (Practice) near the Basal Turn and then decrease.

(2) Partial Withdrawal of the Stylet: As in the SIT, a general increase in net insertion force is observed for the partial stylet withdrawal method. However these forces are reduced after the electrode touches the outer wall around the Basal turn area (Figure 3.1, point 2.). This is due to a decrease in strength near its tip and a change in electrode trajectory as it recoils to follow the ST inner wall, after the stylet is partially withdrawn. Slight peaks are again observed around the 4mm and 9mm marks, with an average peak force at the lateral wall of the Basal turn of 0.057N (Contour) and 0.050N (Practice). Insertion forces increase to 0.041N (Contour) and 0.058N (Practice) prior to the first rib contact. Near full electrode placement, during rib interaction with the model, peak values of 0.115N (Contour) and 0.12N (Practice) are reached, which are less than those produced from the SIT. Average forces associated with stylet removal at partial and full insertion depths are 0.247N and 0.261N for the Contour (0.254N total average).

(3) Advance Off- Stylet Insertions: Notably lower insertion forces result from application of the AOS technique using the Contour Advance electrode. A slight increase occurs during initial insertion (Figure 3.1, point 1.) to an average value of 0.0052N. An increase in frictional force as the electrode slides along the ST inner wall (due to an increase in contact surface area) may contribute to the rise in insertion forces during this period. The force then appears to reduce to a negative value (indicated by the

trough in Figure 3.1, between points 2. and 3.). This corresponds to the change in direction of forces exerted on the inner wall as the electrode pushes against it in the opposite direction to the insertion. Frictional forces continue to increase the magnitude of the total force imparted on the model during this interval. An average peak force of 0.0082N is reached just before the first rib touches the ST (Figure 3.1, just before point 5.). Contact between the marker rib and the model causes a significant increase in output force (represented by the spike in Figure 3.1, point 5.), for the AOS method at a distance of approximately 16 mm along the ST. Here, forces rise to an average value of 0.0504N.

Electrode trajectories appear to vary depending on the technique selected for insertion. Removal of the stylet causes the array to assume a position closer to the modiolus. In the AOS method, the electrode tends to follow the ST inner wall throughout the entire implantation, as shown in Figure 3.1. Insertion studies are performed to see what effect early advancement of the Contour Advance would have on results for the AOS insertion. In these trials, the electrode is inserted 1mm to 3mm into the ST opening, as in the SIT, and the electrode is advanced off the stylet. Out of 30 trials, in 40% of cases the silicone tip rolled back upon itself, precluding deep insertion into the model and prohibiting the tip from assuming a final position along the modiolus, the tip itself pushing the electrodes back towards the outer wall. This increases the chance of inducing trauma in this region, since the contact area is greater in the tip region and may lead to an increase in contact pressure upon collision with the ST wall. The Contour Advance is only meant to be used with an AOS insertion, where the array is inserted to the white marker on the silicone carrier, before the electrode is moved off the stylet.

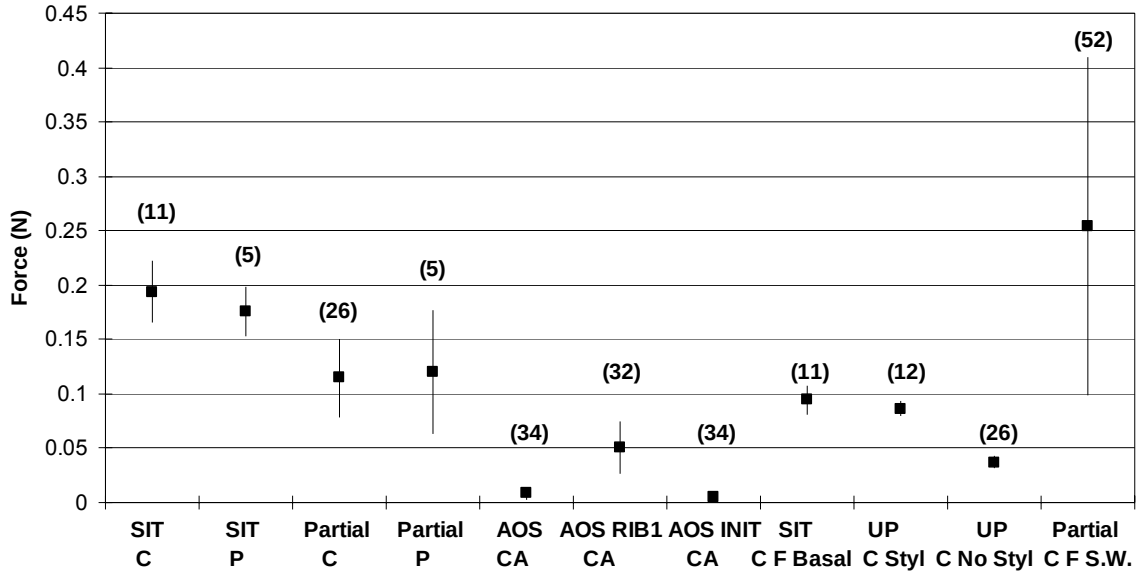

Figure 3.6. Summary of results: average insertion forces with first standard deviation for the Contour Practice (P), Contour 24 (C) and Contour Advance (CA) electrodes for the SIT, Partial withdrawal (Partial) and AOS insertion methods. Also included are peak forces at the Basal Turn for the SIT with Contour array (F Basal), forces due to electrode (UP) removal for stylet (Styl) and no stylet (No Styl) with the Contour, and forces associated with stylet withdrawal (F S.W.) for the Partial withdrawal technique are also provided. Further, for the AOS insertions, the average peak force during initial electrode advancement, which occurs just prior to a change in direction of net forces against the inner wall (INIT) and the average peak force generated when the first marker rib touches the cochleostomy site (RIB1) are documented.

3.6.2 Coefficients of Friction

Results for starting and dynamic coefficient of friction measurements between the Teflon/Silicone interface are summarised in Figure 3.7. The closest representation of the insertion scenario is the silicone tip against a lubricated, rough disc (a Teflon disc that

has been cut via rotational machining and its surface remains unpolished). The profile generated for this representation is shown in Figure 3.8. This set-up gives an average starting coefficient of 0.0605 and a dynamic coefficient of 0.0395, for the time period 1s to 8.5s which is the average time for an electrode insertion (based on the SIT).

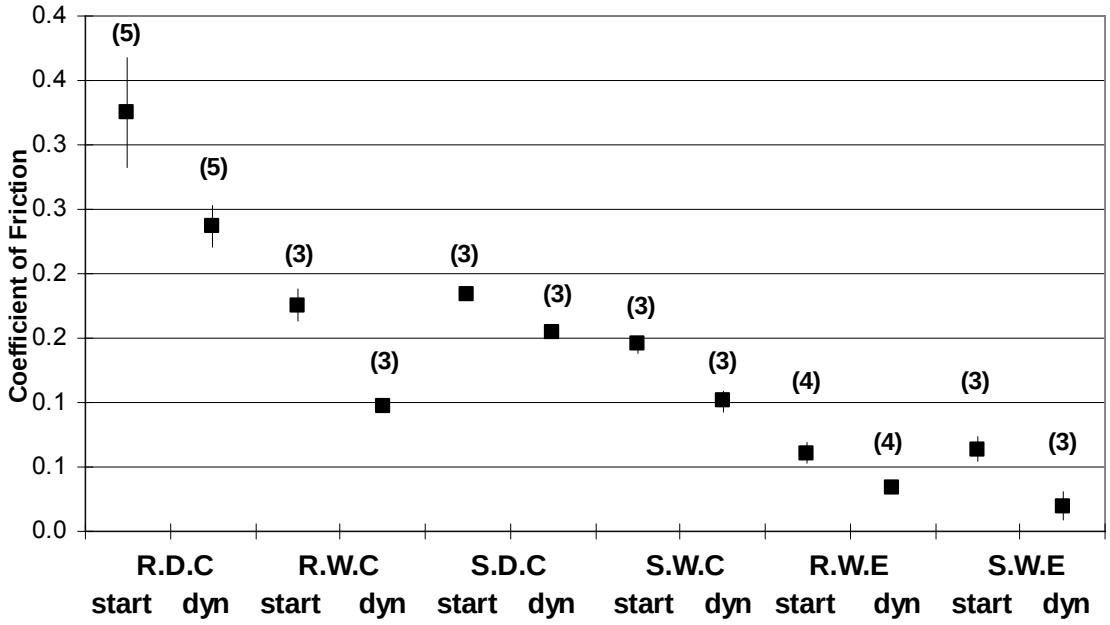

Figure 3.7. Averages and first standard deviations for starting (start) and dynamic (dyn) coefficients of friction between silicone/Teflon interface, for different configurations: rough disc (R), smooth disc (S), dry (D) and wet (W) conditions, with rectangular (C) and spherical (E) silicone geometries.

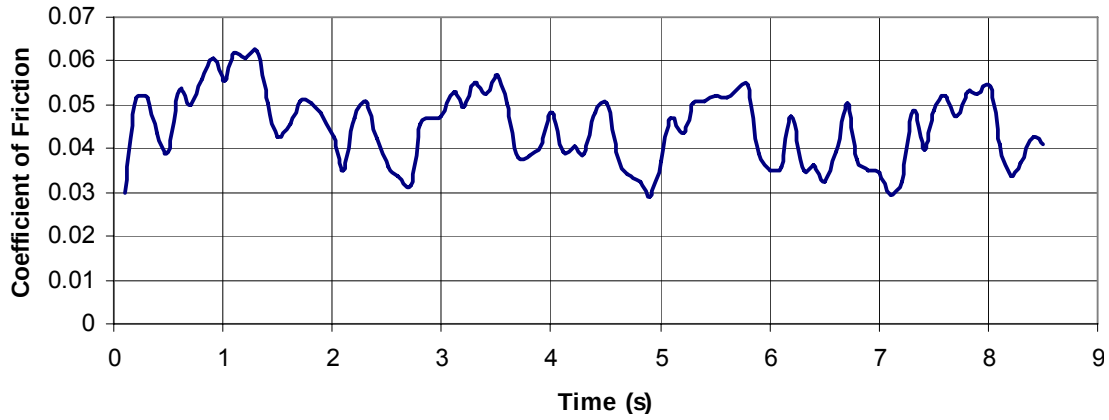

Figure 3.8. A coefficient of friction result for silicone sphere geometry on a rough disc immersed in lubricant (10% bathox, 90% distilled water solution).

Profilometry tests were performed to quantify variation between rough and smooth disc surfaces. Scratch width was set at 5mm and samples were taken at a rate of $0.5mms^{-1}$. Results for each disc are tabulated in Table 3.1. Parameters provided are different measures of surface roughness.

TABLE 3.1. Profilometry test results.

Parameter	Rough (R) (µm)	Smooth (S) (µm)	Diff (R-S) (µm)
Ra	6.2865	0.123	6.1635
Rq	7.9395	0.192	7.7475
Rsk	-0.9475	-3.58	2.6295
Rku	3.6975	30.53	-26.8325
Rp	17.7505	0.546	17.2045
Rv	28.9715	2.056	26.9155
Rt	46.7215	2.602	44.1195
Rc	24.341	1.663	22.678
RSm	0.442	2.363	-1.921
Rdq	13.906	1.471	12.435
Rk	14.823	0.331	14.492
Rpk	4.5775	0.181	4.3965
Rvk	13.518	0.358	13.16

Table 3.1 shows the profilometry test results for a scratch test on a dry Teflon

disc. Measurements are taken for the smooth and rough discs which are used to measure coefficients of friction between the discs and silicone samples. Values include roughness area (Ra), root-mean-square (Rq), skewness (Rsk), kurtosis (Rku), peak roughness (Rp), valley depth (Rv), total roughness (Rt), peak count (Rc), mean peak spacing (RSm), root-mean-square average (Rdq), core roughness (Rk), peaks during run-in period (Rpk), and valleys that retain lubricant in functioning part (Rvk).

Insertion force and coefficient of friction data produced in this work are compared with previously published results for different electrode designs and insertion methods. This is done to validate the results and for comparison of force output and electrode trajectories of the available techniques. The importance of these results is discussed and the key factors that affect force output during electrode insertion are identified.

3.7 Validation of Results

In this section, a critical analysis of the results produced in this work is carried out by comparing them with the previously published data. Force profiles generated from electrode insertion in [35] and [46] are of similar shape to those produced in the current work for the SIT, with insertion forces increasing to a peak value near maximum insertion depth. The Clarion electrode reaches a peak value of approximately 32.5g (0.32N) and the prototype electrode of about 34g (0.34N) [35]. The authors attribute the rise in insertion force to frictional forces associated with electrode carrier positioning. Depending on electrode location (hence proportion of contact surface area between the carrier and ST walls), frictional force will vary. Similar force profiles are generated for the C40+ and C40+ FLEX electrodes [46], reaching average peak values of ~0.036N and ~0.023N respectively. Insertion forces in this work also increase to a peak value

before the first rib touches the model. These values are 0.113N (Contour) and 0.09N (Practice Contour). Removal of the platinum stylet reduces peak values at the same distance to 0.041N (Contour) and 0.058N (Practice Contour). The peak insertion force for the AOS insertion is lowest at 0.0082N (Contour Advance), yet the force increase is not as steep (Figure 3.1).

Insertion forces for all three techniques produced lower insertion forces (both before and after contact with the first marker rib) than those in [35]. However, insertion forces produced in this work are higher than those presented by Adunka et al. [46], for the SIT and partial withdrawal methods. This is most likely due to differences in electrode design and insertion technique. Using the SIT and partial withdrawal method, the presence of the stylet increases the rigidity of the electrode. It does not appear that the carrier used by Adunka et al. [46] houses a stylet. This reduction in strength may account for lower force output.

Insertion forces are lowest for the AOS technique in comparison with results produced by other methods [35, 46]. The force exerted onto the model is 0.0082N prior to interaction with the first rib, whilst insertion of the FLEX electrode gives a peak insertion force of 0.023N [46]. Again, this may be due to differences in electrode properties and methods for insertion. In the AOS insertion, the electrode does not make contact with the outer wall, has a soft-tip to create a softer region in this area and the stylet is not inserted with the carrier which decreases its overall strength. In this work we consider peak forces prior to the point of contact between the first rib and ST. Interaction between the rib and ST opening dominates force output, which does not provide a true representation of force delivery inside the model.

For the SIT, electrode trajectories primarily follow the ST outer wall to full insertion depth, as for the Clarion electrode [35] and as is simulated for the Nucleus

Straight electrode [26]. The electrode tip similarly contacts the lateral outer wall in the Basal turn area [26, 35, 36]. A more medial position is achieved by a new prototype developed by Adunka et al. [35]. For an AOS insertion, the implant follows the ST inner wall and assumes a perimodiolar position at full insertion depth.

Similar work carried out by Rebscher et al. [36] for silicone on epoxy with a 25% soap solution have resulted in a coefficient of friction of 0.6. In this work, a silicone sample is rotated on a Teflon disc in the presence of a lubricant which simulates the suface interactions during electrode insertion into a ST model. The results produced for dynamic and starting coefficients of friction are 0.0605 and 0.0395 respectively, using silicone/Teflon interface with a 10% bathox soap solution. This difference in results may be due to the surface properties of the Teflon itself (including material composition and surface roughness), which gives a much lower coefficient than an epoxy surface for similar lubrication and sample size.

Results produced in this work from insertion force measurements, electrode trajectory and coefficient of friction analysis are comparable with the outcomes of similar research [26, 35, 36, 46]. The SIT is equivalent to insertion methods applied by Rebscher et al. [35] and Adunka et al. [46]. The partial withdrawal method is a modified version of this and the AOS insertion is a newly evaluated technique.

3.8 Discussion of Results

Measurement of insertion forces and coefficient of friction has revealed some critical factors that contribute to force delivery during insertion of the Contour and Contour Advance electrodes into a synthetic replica of the human ST. Examination of the results reveals that carrier strength, contact pressure, frictional force, electrode trajectory and surgical technique each have an impact on force output during an implantation. These

key elements that contribute to insertion force need to be addressed for accurate modelling and validation of the simulation.

3.8.1 Effect of the Platinum Stylet

In this section, the effect of carrier strength due to the stylet and contact pressure on force output during the insertion was examined. Strength of the Contour and Contour Advance electrodes is increased by the presence of the platinum stylet. Results indicate that the greater the stiffness properties of the carrier itself, the higher the total force imparted on the cochlea during implantation, as the related work [26]. Insertion forces associated with the SIT are the highest of all methods (Figure 3.6). For the SIT, the stylet remains in place during electrode advancement. Partial withdrawal of the stylet in the region of the Basal turn leads to a decrease in insertion force, as the carrier strength near the tip of the array is reduced. This is evident in Figure 3.1. After withdrawal around the 9mm mark insertion forces decrease following partial stylet removal after touching the lateral outer wall and then continue to increase. A lesser peak insertion force is reached by partial stylet removal than the SIT, both prior to and following introduction of the first marker rib. Contact pressure at the Basal turn is the same for both techniques, as the stylet remains in the carrier until after this point and contributes to a rise in insertion force at this stage (Figure 3.1, point 2).

Insertion forces for the Practice and Contour electrodes are similar, indicating that the strength properties between designs do not vary significantly. Hence the Practice electrodes provide a good representation of the true array for specialist training. The AOS technique has minimal insertion forces, with the lowest average peak insertion force of all three methods analysed in this work. This method also results in the lowest variability of forces measured between trials. This indicates that the technique is more

consistent than the SIT and partial withdrawal methods which have higher variability.

Advancing the electrode off the stylet greatly reduces its overall strength since the stylet does not provide additional support as the carrier progresses along the cavity. The addition of a soft-tip on the Contour Advance creates a softer region in this area, in comparison to the Contour array tip. Since the electrode does not touch the Basal turn area for the AOS insertion, there is no peak in insertion force in this region (Figure 3.1). Therefore, there is no contact pressure from input force exerted at this stage. To therefore summarise the effect of implant strength on force application: generally, the greater the electrode carrier rigidity coupled with contact pressure at the tip, the higher the force output during an insertion.

3.8.2 Contribution of Frictional Force to Insertion Force

The contribution of force due to friction to the overall force delivery is examined. For all techniques applied in this work it was evident that as the silicone carrier was inserted into the model, the contact area between the carrier and ST increased (as more of it touched the outer and/or inner walls). This lead to an accumulation in force due to friction (2):

$$\tilde{F}_{F\_TOT} = \sum_{i=1}^{N} \tilde{F}_{F\,i} \qquad (2)$$

where F_{F_TOT} is the total force due to friction, F_F is the force due to friction at point i at the cochlea wall and N is the total number of contact points between the silicone and wall.

Frictional force is proportional to the normal force exerted during insertion (1), where the latter is resolved from the total input force by application of (3):

$$\tilde{F}_n = \tilde{F}_{in}\sin\alpha \qquad (3)$$

where F_n is the normal force, F_{in} is the input force and α is the angle between the input force vector and through the point of contact (Figure 3.3).

As the electrode is advanced into the ST using the SIT, partial withdrawal and AOS methods, the increasing component of frictional force contributes to the rise in insertion force magnitude (Figure 3.3). The input force, F_{in}, is greater for the SIT and partial withdrawal methods than the AOS technique. Whilst there is a restoration force exerted onto the ST inner wall as the electrode is advanced using an AOS approach, this is expected to be much smaller than F_{in} exerted by the user. F_{in} can be resolved into two forces: a reaction or normal force, F_n acting against the ST walls and the force that advances the carrier along the ST. F_{in} appears to provide significant contribution in contact pressure at the Basal turn upon first impact with the outer wall, as reflected by the peak in output force (Figure 3.1, point 2).

Whilst the coefficient of friction is assumed to be relatively constant during insertion, tests revealed that starting and dynamic coefficients will vary depending on surface roughness, contact geometry and lubrication (Figure 3.7). For the Teflon/silicone interface, μ increased with surface roughness in dry conditions. However, addition of lubricant (bathox solution) reduced μ and the effect of surface roughness became negligible. Rectangular geometry for the silicone sample resulted in a higher value of μ than the point contact area of the sphere, suggesting that μ may depend on surface contact area (most likely due to adhesion [194]). The value of μ at the start of testing was consistently higher than dynamic μ for all cases. The most applicable representation for the environment inside the model was the silicone sphere geometry, on the rough (not polished) disc in the presence of lubricant, for a dynamic friction coefficient measured from 1s to 8.5s. This is because the sphere geometry most closely matched the silicone tip of the electrode in size and shape, the rough disc was

machined in the same way as the model surface (unfinished) and the soap solution was a substitute for the endosteum lining of the ST. An average insertion took approximately 8.5s. Although the values for μ were small for this scenario, the additive effect of frictional force during an insertion is apparent and does contribute to the final force delivery. The magnitude of frictional force during electrode insertion has not been determined, but could be obtained by using a two-degree-of-freedom (2DOF) load cell to measure F_n during initial electrode advancement and then calculate F_F based on the value of μ (dynamic) as determined in this work. The exact contribution of F_F to the total insertion force could then be established.

3.8.3 Electrode Trajectories and Positioning during Insertion

Electrode trajectory will vary depending on surgeon technique (Figure 3.1). For the SIT, the electrode primarily followed the ST outer wall during the entire insertion. Partial stylet withdrawal around the Basal turn area resulted in the electrode recoiling towards the inner wall and following it until insertion forced the electrode back to the outer wall near its final position. Withdrawal of the stylet at full insertion depth caused the array to be positioned along the inner wall, with electrode orientation towards the modiolus. The Contour Advance followed the inner wall of the cochlea for AOS insertions, as restoration forces caused the silicone to curl towards it. This eliminated contact pressure between the electrode tip and ST outer lateral wall, which caused a significant peak in insertion forces for both the SIT and partial withdrawal methods (Figure 3.1, point 2). It also reduces the chance of the electrode deflecting upwards and into the delicate BM during an in vivo implantation. This may occur in the SIT and Partial withdrawal after the tip contacts the outer wall at the Basal turn [39]. Final removal of the stylet at full insertion depth caused the electrode to advance slightly, providing a deeper insertion.

It is worth noting that if the electrode orientation is correctly pointed towards the modiolus, there will be a reduced contact surface area between the carrier/wall interface for the AOS insertion as opposed to the SIT and Partial withdrawal methods. Exposure of the electrodes for the Contour and Contour Advance means that they do not come into contact with the cochlear walls and so the total contact surface area on this face is reduced. Since the electrode followed the inner wall only for the AOS and primarily the outer wall for the SIT, F_{F_TOT} was reduced (2), which caused a decrease in insertion force.

3.8.4 Summary of Surgeon Technique

Surgeon technique and selection of electrode type will vary between specialists, however insertion studies performed in this work reveal that this decision will affect force administration and electrode trajectory. The Contour Advance is designed for use with an AOS insertion, which collectively achieve a more desirable outcome than the Contour electrode inserted in a SIT or combined with partial withdrawal of the stylet. An overall reduction in the rigidity of the electrode as it is inserted into the ST, combines with improvement in trajectory as it traces a path along the inner wall, to give a marked reduction in force application, particularly at the Basal turn where previous designs and administration have caused damage or trauma in this area [35-37, 39].

For insertion of the Contour Advance, it is recommended that the array is inserted to the white marker on its carrier, as intended by the manufacturer, in order to achieve optimal placement of the electrode array as well as preventing its damage. Results indicate that early advancement of the carrier off the stylet may result in the tip rolling back upon itself. At worse case this prohibited insertion during trials, causing the array to buckle and in practice it would be discarded. In most instances, bending of the

tip results in a continued insertion, yet at final position the tip remains bent at 180º and pushes the electrode array further away from the inner wall. This would result in an electrode position that is closer to the delicate structures residing along the lateral wall. As a larger tip cross-section is advanced in this region (due to bending of the tip), there is an increased risk of inducing trauma in this area caused by incorrect force application and/or undesirable electrode trajectory.

3.9 Overview of Analysis

For any prosthetic implantation it is important to evaluate the products and associated procedures that the surgeon has available for use. This is essential for accurate modelling of device insertions, including cochlear implantation. It is necessary to investigate the factors contributing to force delivery during an insertion, including electrode design, trajectory and administration technique. Different electrode designs and insertion schemes will affect force administration during surgery, the extent of induced trauma and success of the outcome. In this work, force application is evaluated for implantation of the Practice, Contour and Contour Advance electrodes into a synthetic model of the human ST, using the SIT, Partial withdrawal and AOS methods. This work was performed to investigate force delivery for these techniques and identify factors which affect force administration during cochlear implantation. These elements will affect force delivery during insertion and must be identified in order to accurately model the physical behaviour of the implant and its environment. Previous studies have examined electrode trajectories and insertion forces for the Med-El Combi 40+, C40+ FLEX, Clarion and custom designs.

Results produced in this work indicate that forces imparted on the ST during an insertion are dependent on electrode strength, trajectory and frictional forces between

the silicone/wall interfaces. Higher electrode strength combined with contact pressure between the tip at the Basal turn and accumulation of force due to friction increases total forces associated with cochlear implantation. Minimal force application was achieved using the Contour Advance electrode in an AOS insertion. This method prevents the electrode from touching the ST lateral outer wall in the Basal turn area, which is expected to eliminate contact pressure and minimise trauma in this region. In order to avoid buckling of the silicone soft-tip, the Contour Advance should be inserted to the white marker, as intended by the manufacturer. Early advance of the electrode may prohibit both its optimal placement along the ST inner wall as well as a deeper insertion. Results from this study will be used to validate a surgical simulator with force-feedback for training surgeons in cochlear implantation. Research of this kind will assist in improvement of administration techniques for cochlear implantation.

3.10 Selection of SIT for Simulation

The insertion technique chosen for the simulation is the SIT, with the Contour electrode design. The partial withdrawal method is a modification of the SIT and the SIT may be modified to include a partial stylet withdrawal around the Basal turn region as an extension of the work. Whilst the AOS insertion gives improved results in terms of force output onto the ST walls during cochlear implantation, it is a relatively new procedure and in practice, surgeons are still using either the SIT or partial withdrawal methods. Therefore it is necessary to replicate the SIT in a virtual environment for teaching purposes. By practicing the SIT on a surgical simulator, it is envisaged that the surgeon may improve his or her administration technique and be able to establish the extent of force application that is appropriate for this type of insertion. For simulator development, the SIT may first be modelled for insertion of the Contour electrode and

this same design could then be modified to include a partial stylet withdrawal. Future designs could then model the unique AOS insertion of the newly developed Contour Advance electrode.

Visual and haptic rendering of an electrode insertion will be aimed at replicating the implantation of the Contour electrode using the SIT. Simulator design will consider the analysis of electrode design, behaviour and administration technique presented in this review. Insertion force results and coefficient of friction measurements will be used to model the physical interactions between the ST and implant. Finally, in the validation section of work (Chapter 6), the insertion force profiles generated by insertion of the Contour electrode using the SIT will be compared with results produced from the simulation. This will provide a quantitative measure of simulator accuracy.

3.11 Limitations of the Experimentation

The two-dimensional, synthetic model of the ST used in the experimentation has shortcomings. The use of a three-dimensional model of the ST from Teflon is preferred, as the actual cochlear spiral is three-dimensional and as the electrode progresses from base to more apical regions it will travel in all three dimensions. Use of cadaver material for insertion analysis is recommended for comparison of results obtained using the two-dimensional replica. To date, force measurement studies have used synthetic models of the ST as technology enabling in vivo measurement is presently not available for this surgical application. It is difficult to replicate tissue properties and quantify insertion forces that directly inflict trauma in a plastic model of the ST, which is not a true scenario of the actual procedure. The Teflon model is of different material composition and physical behaviour than a live human cochlea. In this work, a 1DOF Instron device is used for measurement of insertion force, which does not capture all force components

contributing to the total insertion force. Specifically, measurement of F_n would enable the quantification of F_F. Ideally, a 6DOF force measurement device should be used.

Chapter 4

GEOMETRIC MODELLING OF THE HUMAN COCHLEA

4.1 Introduction

A three-dimensional replica of the human ST is first derived from CT and then from measured data, for the purpose of a virtual cochlear implantation. The Nucleus® 24 Contour™ electrode is replicated for the insertion. Visualisation and force rendering during user interaction with the model is realised using specialised software and commercially available haptic interfacing. The user will be able to perform real-time implantation, using the SIT, into the ST with a haptic device to manipulate the environment and deliver the forces associated with the insertion back to the user. Force, torque and position information will be displayed at the GUI and the generated data is logged during the process to monitor force profiles. Insertion force studies will be performed using an anatomically accurate model of the ST and Nucleus® 24 Contour™ electrode arrays to provide experimental validation of the output from the simulation. This system will provide objective assessment of surgeon technique during electrode insertion into a model of the human ST with the application to train specialists in the clinical procedure.

The system should allow for an arbitrary insertion, including variation in angle of approach, insertion speeds and electrode positioning. The model should incorporate parameters which can be varied to capture changes in patient cochlear morphology, that

is, creation of a patient-specific model. Physical attributes obtained from measured data (such as the coefficient of friction between the implant and ST walls) will be included in the haptic representation of the model. Finally, insertion study results that have been obtained experimentally will be statistically compared with force output from the simulation, in order to validate the surgical simulator.

4.2 Stages for Simulator Development

Geometric modelling is the first stage in simulator construction and this chapter is focused on the design of an anatomically accurate geometric model of the human ST for virtual cochlear implantation. Design of the Nucleus® 24 Contour^{TM} electrode array is discussed in brief. Following geometric model construction and visualisation, dynamic interactions with the model are realised via visual and haptic rendering of the insertion process. This process captures real-time topological changes and delivers forces back to the user during electrode/ST interactions. System evaluation is the final process in simulator development, where experimental and/or subjective validation is provided. In this work, experimentation is performed via insertion force studies to objectively assess simulator results. An overview of the stages for simulator development is illustrated in Figure 4.1.

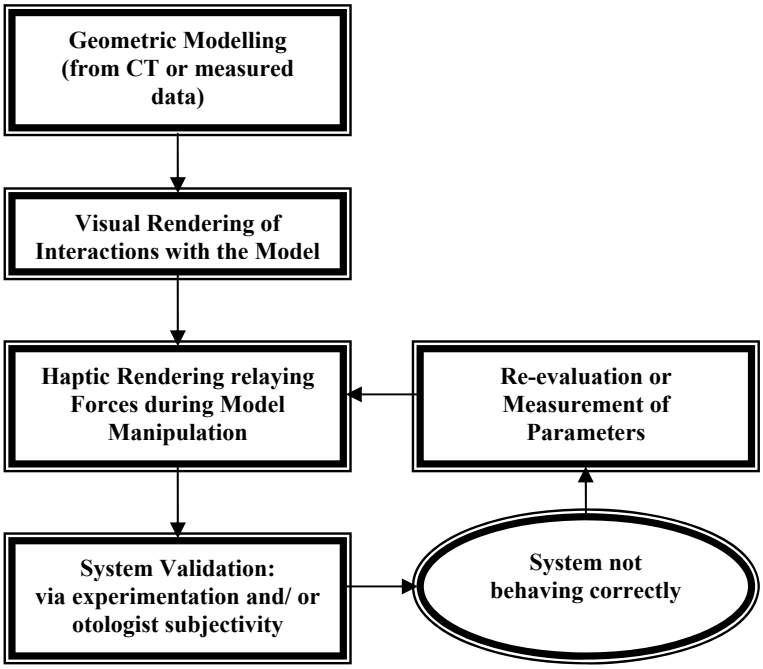

Figure 4.1. Diagram showing the stages of simulator development including model design, implementation and system validation.

4.3 Existing Work

In Chapter 2, the published literature on three-dimensional recreation of the cochlear spiral in a virtual environment was assessed. Significant detail was presented on its reconstruction from imaging modalities including histology [6, 9, 10, 50-53, 55-58, 60, 64, 65, 67], MRI [5, 82] and CT [5, 7, 8, 55, 68-70, 72, 73, 76]. Segmentation techniques associated with manual [9, 10, 50-53, 57, 58, 60, 65, 71, 76], semi-automatic [5, 7, 62, 67-72, 76, 82] and automatic [52, 76] extraction of the cochlea from the temporal bone region were presented. Surface-based [5-7, 9, 10, 50, 53, 56, 57, 64, 65, 67, 69-71, 76, 82] approaches for visualisation of the structure were discussed, as well as volume-based [5, 68, 94, 95] techniques. Representations of the ST and SV as separate passages were also reviewed [5, 82, 91, 92, 95]. In this section, attention is

focused on three-dimensional models of the cochlea produced from measured data as well as mathematical descriptions of the structure.

The cochlear structure is notably small, anatomically complex and differs in tilt and length between individuals. This makes it hard to model precisely and on an individual basis. It also makes the process of parameter selection and extraction a difficult one. Despite its miniature size, complex configuration and variation in morphology, there is a relatively large amount of published work on the measurement of the cochlea. Researchers have taken images [3, 7, 8, 11, 36, 52], casts [11, 35, 36, 52, 195, 196], harvested cadavers [9, 10, 20, 22, 35, 52, 53, 195, 196] and produced three-dimensional reconstructions [9, 10, 35, 36, 52, 53] for this purpose. Average ST inner wall (IW) and outer wall (OW) length [9, 11, 52] (as a function of radial displacement [9, 11, 52]), cross-sectional width [22, 36, 195, 196], height [36, 195, 196] and area [195], SV width [22, 196] and height [196], cochlear canal length [3, 8, 10], radii [8], height [3, 8] and number of turns [3, 9, 10], ST surface area [20], SV/SM surface area [20], RW position [11, 52, 53] and BM/OC length [9, 55] are some of the most common measurements taken within this region. Geometric measurements such as these have been used to form three-dimensional depictions of the cochlea [7, 26, 69, 91, 92, 95] and ST [35, 91, 92, 95].

Rebscher et al. [35] produced casts of the human ST from a synthetic polymer (polymethyl methacrylate) using temporal bone cadavers. Measurements (including ST cross-section width and height) of these casts were then taken to produce three-dimensional virtual reconstructions of the ST. The authors use this information as a reference for insertion force studies [35]. Yoo et al. [7] derive parameters from CT images to produce a mathematical description of the centroid of the cochlear spiral, for three-dimensional modelling of the structure. This is compared with measured data from

Kawano et al. [9]. Geometric measurements, including cochlear height, radial displacement and cross-sectional area, are used to produce another approximation of the cochlea [69]. Hanekom [95] creates a three-dimensional FEM of the ST using previously published geometric data and image data, and then compares this with measurements from existing work. Three-dimensional models of the ST, SV and BM are produced from physical measurements, for the computational modelling of cochlear mechanics [91, 92]. Chen et al. [26] produce a two-dimensional model of the ST for FEM analysis of contact pressures and stiffness during CI insertion. ST cross-sectional width, height and area data from Hatsushika et al. [195] are used to create the FEM [26].

Mathematical models have been produced to approximate the cochlear spiral [7, 8, 55, 69]. Ketten et al. [8] measure cochlear length, which is defined as the centre of the combined SV, SM and ST space (since the CP is not clearly distinguished in CT images [8]), cochlear radii and cochlear axial height. The authors [8] apply this information to fit an Archimedian spiral to the midline of the cochlea for its representation, based on patient-specific data. The Archimedian spiral (4) is first suggested by Ketten and Wartzok [55] as one of the two best approximations for curve representation of the cochlea (the other one being a logarithmic spiral (5)) and these are mathematically defined as:

$$\mathbf{r} = \boldsymbol{a}\,\Theta, \qquad N < 2, \qquad (r_n/n) >= ((r_n + 1)/ (n + 1)) \tag{4}$$

$$\mathbf{r} = \mathbf{e}^{\boldsymbol{a}\,\Theta}, \qquad N > 2, \qquad (r_n/n) <= ((r_n + 1)/ (n + 1)) \tag{5}$$

where r is the radius at cochlear turn number n, N is the number of cochlear turns, Θ is the angular distance along the spiral in radians and *a* is the spiral size constant [55].

Yoo et al. [7] model the central path of the cochlear canal as a single chamber, by a helico-spiral approximation, which is a generalisation of the Archimedian spiral. The helico-spiral is defined by the following expression (6).

$$\mathbf{R} = a\, \mathbf{e}^{\,b\,\Theta}, \qquad \mathbf{h} = c\, \mathbf{e}^{\,d\,\Theta} \tag{6}$$

where R is the cochlear radius at angular position Θ (in radians) as the cochlea spirals from base to apex and h is the cochlear height at angle Θ. Constants *a*, *b*, *c* and *d* are derived for patient-specific representations, from CT using parametric derivation processes as described in the literature [7].

Three-dimensional surface representations are fitted to the spiral [7]. Surface depictions are also produced by segmentation from CT using Analyze and from measured data [9] for model comparisons [7]. The helico-spiral is a logarithmic estimation of the central spiral of the human cochlea and the authors [7] reveal that, compared with measured data [9] defining the same path, error increases towards the base of the cochlea since it is not truly represented by a logarithmic spiral in this region but tends to flatten out. This scenario is rectified in the work [69], where the central path is instead represented by an extension of the template derived by Cohen et al. [11]. In the new approximation, equations (5) and (6) are extended to include a third dimension; a height parameter, z:

$$\mathbf{z} = \mathbf{e}\,(\Theta - \Theta_1) \tag{7}$$

where Θ is the angular distance in degrees, e is a constant and Θ_1 is the starting location at the RW (10.3º). The value e is determined by using the value 2.75mm as the cochlear height [69]. This gives a constant height increase for the cochlea as it spirals towards the most apical point (910.3º) [69].

Results show an improvement at the base of the cochlear spiral in comparison to the helico-spiral approximation and more closely resemble a real model of the human cochlea derived from CT, in this region [69]. Again, parameter estimation makes the model patient-specific. The three-dimensional model produced by adaptation of this central path definition is used for CI insertion on the internet, yet the insertion is along

this path and not into the ST, as specified by the authors [69] and previously discussed in Chapter 2.

The trajectory of an electrode array is estimated [11, 52] by analysing radiographic images of an implanted array and fitting the following template:

$$\mathbf{R} = A\, \mathbf{e}^{-B\Theta}, \qquad \Theta >= 100^{o} \tag{8}$$

$$\mathbf{R} = C\,[1 - D \log_e (\Theta - \Theta_0)], \qquad \Theta < 100^{o} \tag{9}$$

where A, B, C, D and Θ_0 are constants (optimal values are determined in [52]), R is the radius from the modiolar axis and Θ is the angular distance along the spiral in degrees [11, 52]. This template is compared to measurements of the ST outer and inner walls as a function of radial displacement [11, 52], however this trajectory is in two dimensions and does not consider cochlear height as the spiral progresses from base to apex.

4.4 Model Development from CT

In this work, a first approximation of the cochlear spiral is produced from CT for virtual CI insertion. Upon image acquisition from spiral CT and registration onto the PC, the image sequence is processed using the Analyze software. The temporal bone volume is segmented from the image series using a semi-automatic threshold-based approach. VTK is then employed for surface rendering of the cochlea. The three-dimensional structure is assessed by an experienced surgeon, for clinical validation.

4.4.1 Image Acquisition and Registration

Spiral CT scans of a right temporal bone cadaver were taken with a Siemens Somatom Plus 4 scanner, at the Royal Victorian Eye and Ear Hospital (RVEEH), in Melbourne, Australia. The human cadaver was acquired from the institute's temporal bone drilling laboratory and selected for scanning by an experienced ENT surgeon, who examined

the specimen and deemed it to be in good condition (no noticeable degeneration caused by disease or other factors), intact and in its entirety (no structures missing). Upon inspection, skin, muscle, nerves, an artery and vertebrae could be distinguished. An imaging technician positioned the cadaver and scanned it from superior to inferior, anterior to posterior. Cross-sectional resolution was 512 (pixels/centimetre) by 512. Initially, spacing between image planes was 1mm and slices were later reconstructed to provide 0.5mm spacing between each scan. The image sequence was transferred from raw to DICOM data format in a lossless transfer.

4.4.2 Image Processing and Segmentation in Analyze

Analyze is a biomedical image processing and visualisation application that supports a variety of image modalities, including CT. Image sequences may be viewed, manipulated and measured interactively by the user. A range of functions are available for feature enhancement and segmentation that use either manual or semi-automatic processes. Various interpolation and shading methods may be selected for volume rendering techniques. In this work, modules that were used for image processing include: *Import/Export*, *Image Edit*, *Morphology*, *Image Calculator* and *Volume Renderer*. These are all semi-automatic processes that require some degree of user input.

Following image acquisition via CT scanning and registration onto the PC in DICOM format, the sequence is imported into Analyze for data processing. Scan files are sorted automatically prior to import, to ensure that the sequence is in chronological order. This is important, since scan sequences can be stored or transferred out of sequence and this will invariably create erroneous data upon volume generation. Once loaded as a single volume, anisotropic voxels comprising the image sequence are forced to cubic geometry using linear interpolation. Following import into Analyze, individual

scans within the volume are viewed using the *Image Edit* module. With this functionality, anatomical landmarks such as the cochlear first and second turns, are identified by visual comparison with published work [197].

To successfully segment the temporal bone region, including the cochlea, thresholding is applied. This method is chosen since bone is of high density and often has specific grey levels, which exhibits good contrast with softer tissues captured by CT scans [198]. A semi-automatic threshold-based method is effectively used by Pflesser et al. [74] to segment thirty objects within CT scans. It should be noted that certain structures still require manual segmentation, including the facial nerve and auditory ossicles [74]. In this work, thresholding is applied using a semi-automatic approach, within the *Morphology* module, to segment the structures comprising the temporal bone. Threshold limits are interactively changed using an interface that facilitated effective and fast user interaction. As these boundary values are varied, a binary slide sequence is updated on the display, showing voxels (in white) whose intensities are within this range.

The temporal bone region is comprised of four main constituents: the petrous, mastoid, squamous and tympanic bone sections. These four structures are preserved and extraneous information eliminated using the threshold approach. An intensity range of 300 to 5904 is selected to satisfy this requirement. Again, published works [197] are used to cross-reference anatomical landmarks identified in the images, to ensure that no information is missing from the processed image. Model verification is carried out using a sub-function of the *Morphology* module, the *Step Editor*, as well as inspection by an experienced ENT surgeon. As an output from this stage, a binary volume, formed from processed image scans of the temporal bone, is produced. Figure 4.2 shows an original

image imported into Analyze and Figure 4.3 is a processed image of the temporal bone, both revealing the cochlea (identified by an arrow).

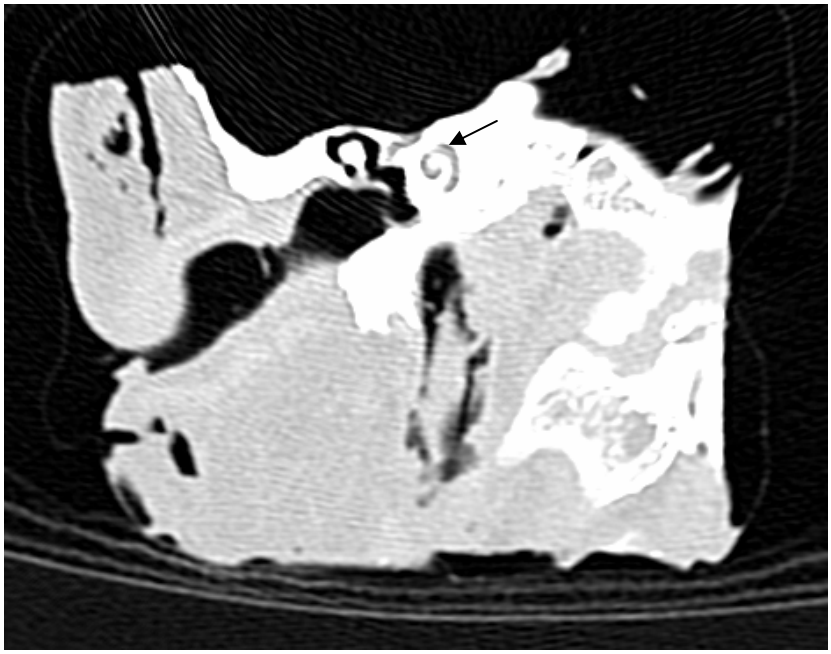

Figure 4.2. An original image slice of the temporal bone in Analyze.

A binary dataset is produced from the method previously described. To derive a subset of the original temporal bone volume that consists of voxels belonging only to structures of interest within this region, a matrix multiplication is carried out. The *Image Calculator* module provides a quick, easy multiplication of the two arrays: the original grey-scale volume of the temporal bone and the binary volume. The resultant volume is displayed using *Volume Renderer*, which provides a three-dimensional perspective with *Gradient Shading* as the render type. *Volume Renderer* functionality allows for interactive display and rotation of the derived volume. Upon rotating the volume about each of its three axes, artefacts or high intensity noise surrounding the temporal bone segments are observed. Initially, artifacts are removed manually using *Image Edit*.

Unwanted regions in each scan are manually identified, outlined and deleted. This proved to be a laborious task that could not easily nor quickly be replicated. A second, semi-automatic method was then chosen for this purpose, using *Slice Edit* in *Volume Renderer*. This also proved arduous and a final method for fast elimination of artifacts was instead selected. The latter makes use of the *Connect Tool* in *Volume Renderer*. Again this is a semi-automatic process that applies a seeded region grow, with 26-voxel connectivity. The user selects a single seed pixel which is positioned on the mastoid portion of the temporal bone (wherein the cochlea resides). Once the connection process is initiated, voxels connected to the seed pixel are kept as part of the Region of Interest (ROI). Successful elimination of surrounding artefacts is achieved using the connected-components method, which isolates the temporal bone from peripheral bony structures (Figure 4.3).

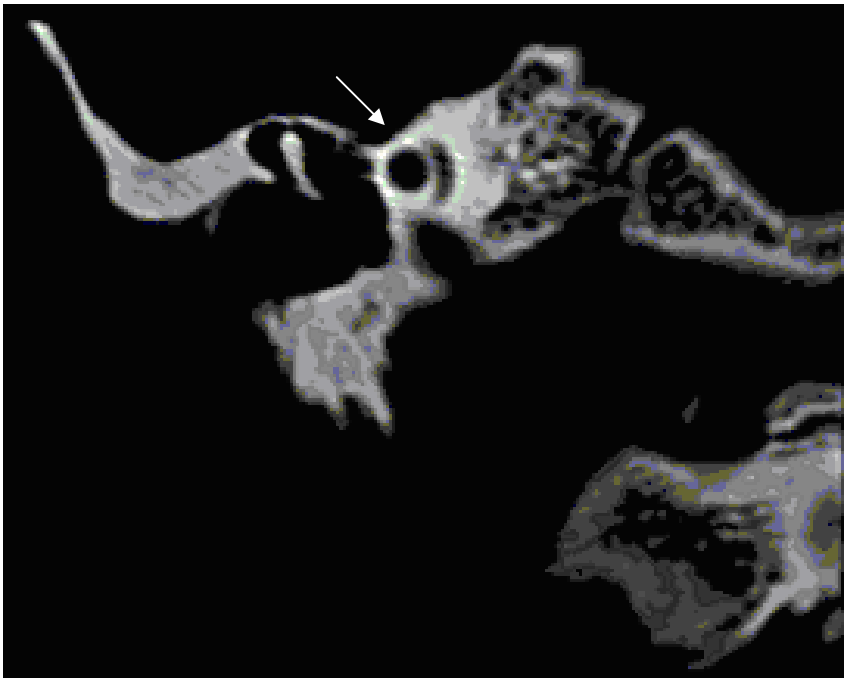

Figure 4.3. A processed image of the temporal bone scan slice, including vertebrae and cheekbone components.

4.4.3 Three-Dimensional Reconstruction in Analyze

Interactive rotation and display functionality in *Volume Renderer* is again used to view the volume produced after elimination of noise surrounding the temporal bone region. An ENT specialist examined the virtual volume and identified inclusion of unnecessary structures: the vertebrae and cheek bone components (Figure 4.3). Such information is not required for volume visualisation of the cochlea. Another sub-function within *Volume Renderer*, *Object Separator*, is effectively used to remove this undesirable data. To initiate the semi-automatic segmentation process, a seed pixel is selected on both sides of the object boundary to identify the regions to be separated. This is done for both the cheek bone and vertebrae structures. Clearly defined, natural edges exist around these components, which make the segmentation process uncomplicated. An ENT surgeon again examined the virtual specimen, using both interactive rotation capabilities offered by *Volume Renderer* to view a three-dimensional perspective and analyse individual, two-dimensional image slices with *Edit Review* in the *Image Edit* module. Extraneous data are effectively eliminated using object separation techniques available in Analyze. By applying a Marching Cubes algorithm available within the *Surface Extraction* module, a surface description of the volume is generated (Figure 4.4). The volume file wrapper function in Analyze is used to create a series of two dimensional files in Tagged Image File Format (TIFF) format (one slice is shown in Figure 4.5). The final volume itself is stored in the custom AVW format.

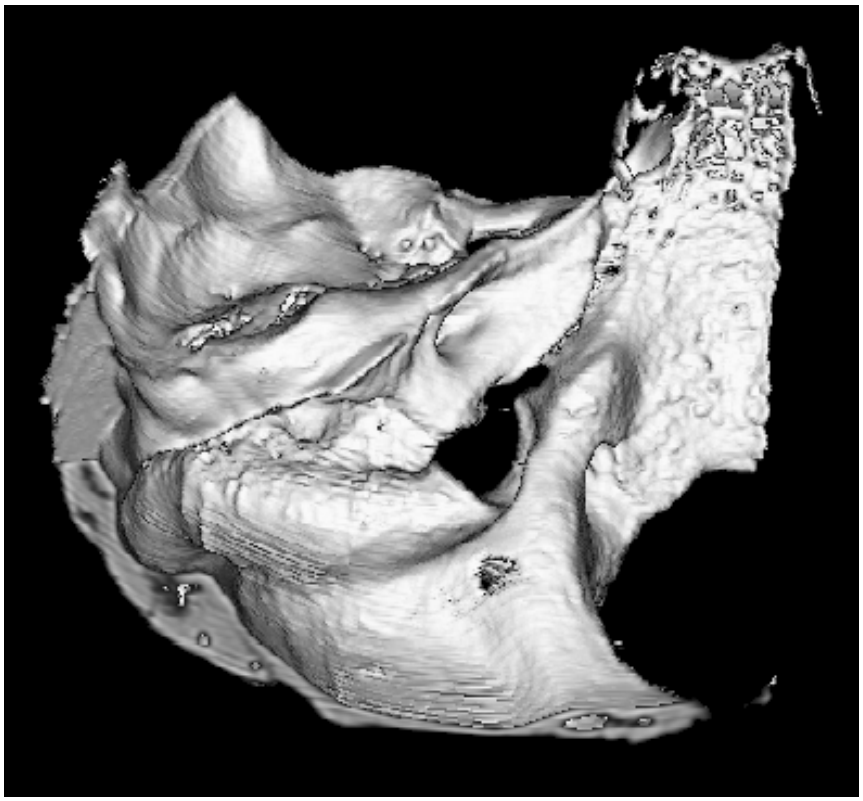

Figure 4.4. Human temporal bone volume produced in Analyze.

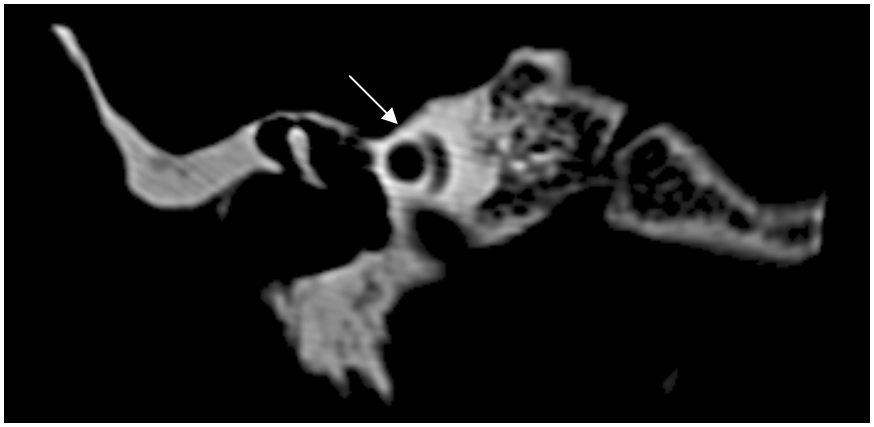

Figure 4.5. Two-dimensional image slice from temporal bone volume.

4.4.4 Surface Rendering of the Cochlea

A program sequence is written (Appendix A) using the Visualization Toolkit (VTK) and Python to provide a GUI that enables semi-automatic extraction of the cochlea from the surrounding temporal bone volume produced with Analyze. There are four algorithms in the extraction series, which are executed from the command prompt via:

>> python *filename.py*

The first program, *Visualise.py*, imports the temporal bone volume in AVW format, crops, scales and outputs the volume as a series of images in the VTK GUI for review. Next, a new image sequence of the ROI is created by running *Binary.py*. During program execution, the user can specify an upper threshold (256 selected) and lower threshold (66 selected) for segmentation of the cochlea from extraneous data that may still exist and render the image over this threshold. Implementation of the segmentation stage is performed in *Segment.py*, which generates a segmentation window (Figure 4.6) that shows the current image slice for ROI segmentation and a GUI (Figure 4.7) that provides user-selectable parameters for feature extraction.

Figure 4.6. The Segmentation Window for interactive user semi-automatic threshold-based feature extraction of the cochlea (denoted by an arrow).

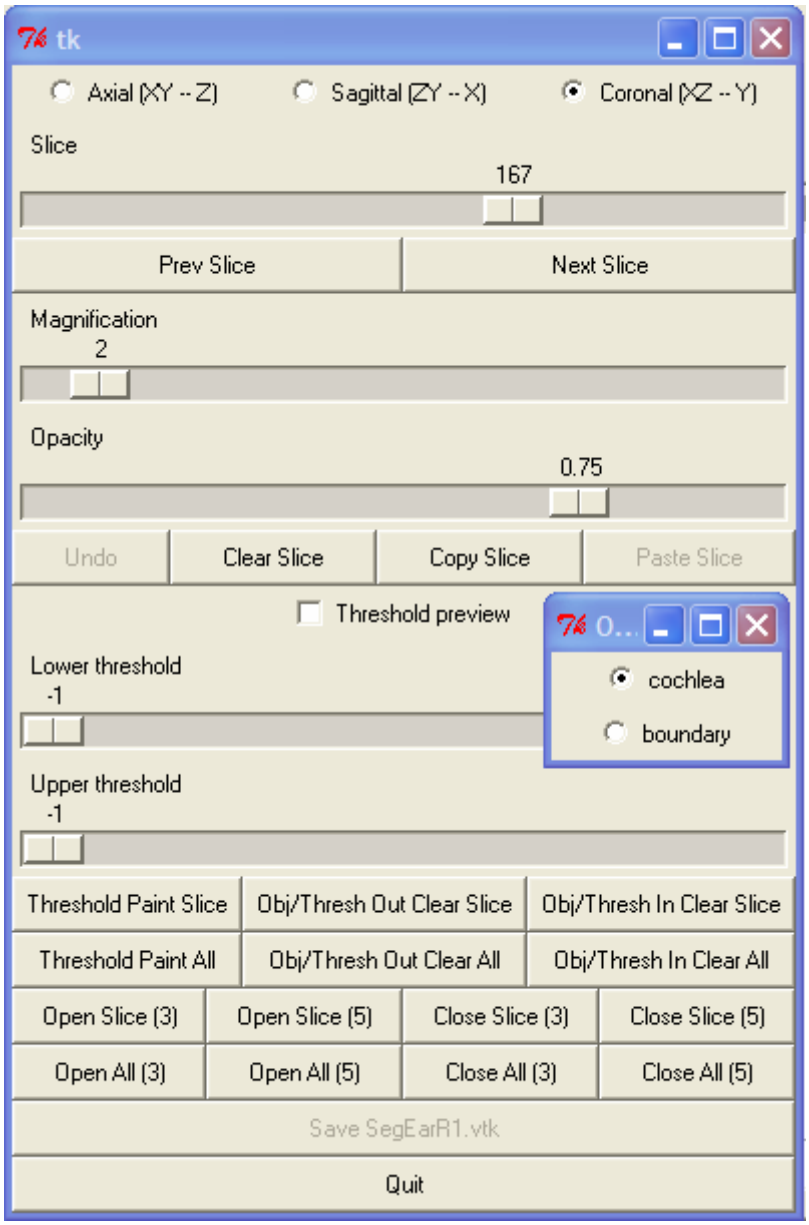

Figure 4.7. The GUI for surface extraction of the cochlea from 2D images, generated using VTK and Python.

To effectively segment the area using a threshold-based approach, the following steps are executed:

1. *boundary* is selected
2. *threshold preview* is chosen

3. set upper threshold to 256, lower threshold to 66
4. select *Threshold Paint All*

In effect, all pixels within this region, throughout the image series are selected.

The specific region of the cochlea is defined using a semi-automatic edge-drawing method:

1. select *cochlea*
2. manually draw the edges around the cochlea in the Segmentation Window, using the cursor. The edges must form a closed-in area: this is the ROI.
3. hold the shift key, then position the cursor within the ROI and left click with the mouse button. This selects all pixels within this region.

By repeating steps 1 to 3 for the semi-automatic edge definition of the cochlea, for the image slices in which the cochlea appears the entire cochlear structure is segmented. The binary image produced via thresholding is combined with the ROI defined by the user to give a final two-dimensional image sequence of the cochlear region.

The final program *Render.py* generates a VRML surface description of the selected region (cochlea) which is super-imposed on a two-dimensional cross-section of the volume (Figure 4.8). At the GUI, the user can select which slice in the image sequence is displayed, via a scroll bar, and rotate or re-size the scene to any angle in three dimensions.

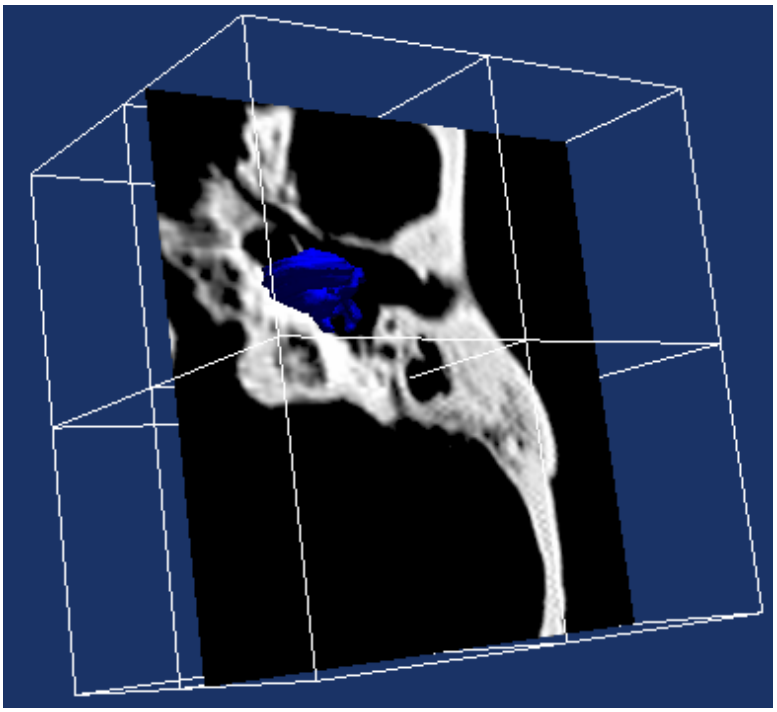

Figure 4.8. A three-dimensional model of the cochlea constructed in VTK. An image slice is shown super-imposed on the cochlea structure (blue).

4.4.5 Model Validation and Shortcomings

The VRML script defining the segmented cochlea is modified in accordance with applicable Reachin API node definitions. It is visually as well as haptically rendered in the Reachin API (refer Figure 4.9 for graphical depiction and Appendix B for code). The latter is achieved by adding a *SimpleSurface* node to the VRML script defining the cochlea. The program is executed in the Reachin API by running the following command:

>> Reachinload *filename.wrl*

The user can rotate and translate the virtual cochlea surface to assess the structure. A single, inner cavity of the cochlear spiral could be felt during haptic interaction with the model, indicating that the complex array of tissues (CP) which separates the ST from

the SV had been inadvertently discarded during the segmentation stage in Analyze. Further, an ENT specialist assessed the dataset during interactive segmentation in VTK and found that some of the cochlear structure was missing. It was supposed that this data was eliminated by initial thresholding of the CT scans during temporal bone segmentation, being of lower density than the accepted ROI. CT has been used previously for three-dimensional model construction of the cochlear [71], yet data processing of the image sequence can lead to elimination of the CP. This leaves the cochlea inaccurately represented as a single canal. It is an important criterion for the individual chambers inside the cochlea to maintain their integrity during image processing or formation of the model, for CI insertion into the ST. It was determined upon discussion with the ENT specialist that in order to produce separate cavities for the ST and SV for the purpose of CI insertion, another technique must be used.

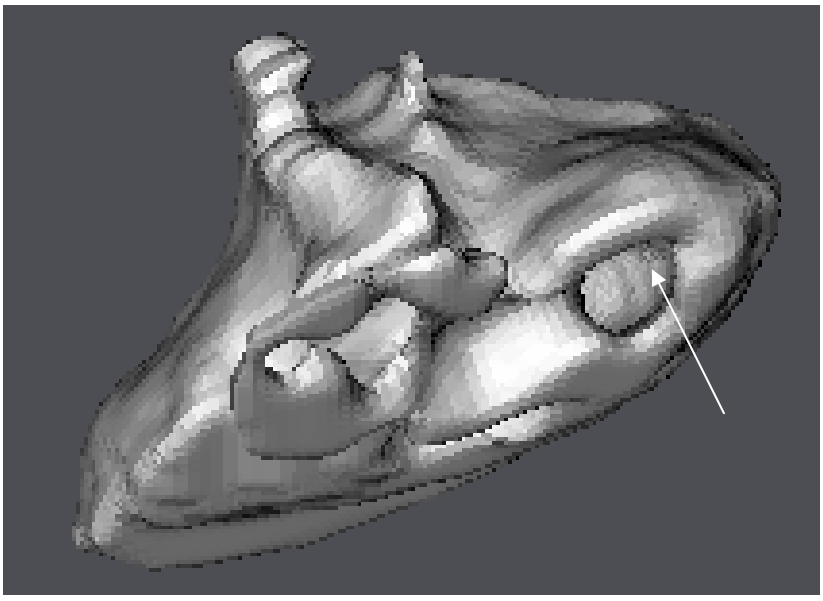

Figure 4.9. A surface representation of the cochlea which is visually and haptically rendered in the Reachin API. The cochlear opening near the site of the RW is observable (denoted by an arrow), which reveals a single chamber.

4.5 Model Development from Measured Data

A parametric model of the human ST has been derived from measured data [3, 8, 9, 11, 92, 195] for the haptic-rendered simulation of CI insertion. This section of work describes the method and results associated with three-dimensional model development. Model design and techniques for parameterisation are presented, as well as FEM surface construction in ANSYS. The final reconstruction is compared with two existing descriptions of the cochlear spiral, produced from mathematical approximations [7, 8]. A first approximation of the BM is also produced. Finally, electrode array design and its synthesis in ANSYS are briefly discussed.

4.5.1 ST Model Design

In the ST model design phase, existing publications detailing various ST physical measurements were analysed to determine the most appropriate sources for construction and parameterisation of the model. From this selection of literature, the geometric data used for model design, realisation and validation included:

- ST width, height and cross-sectional area [195]
- ST IW and OW measurements at each ¼ turn [9, 11]
- OC length [9]
- Radial lengths at each ¼ turn about the modiolar axis [11]
- Cochlear axial height [3, 8]
- Cochlear diameters at each ½ turn [3, 8]
- BM width variation [92]

First, the data provided by Hatsushika et al. [195] is evaluated. Whilst ST height, width and cross-sectional area are given as a function of ST length from the RW, there is no information relating to angular displacement or change in cochlear axial height.

Further, the data only covers approximately 70% of the ST length and so extrapolation of the data is required. The data [195] must be combined with measurements (ST IW and OW length, radii at each quarter turn, cochlear height) as a function of angular displacement. Consequently, the measurements for IW and OW radii are plotted at each ¼ turn to 2 ¼ turns [11]. This information [11] was also employed for construction of the ST model used in the insertion force analysis and experimentation detailed in Chapter 3. IW and OW lengths are plotted on the same diagram to each ¼ turn until just prior to 2 ¾ turns (2.63 turns) [9]. This information is plotted in a two-dimensional plane (Figure 4.10). Percentages of progressive IW length of its total length and OW length of its final value are also noted [9]. The measurements used for spiral construction are given in Table 4.1.

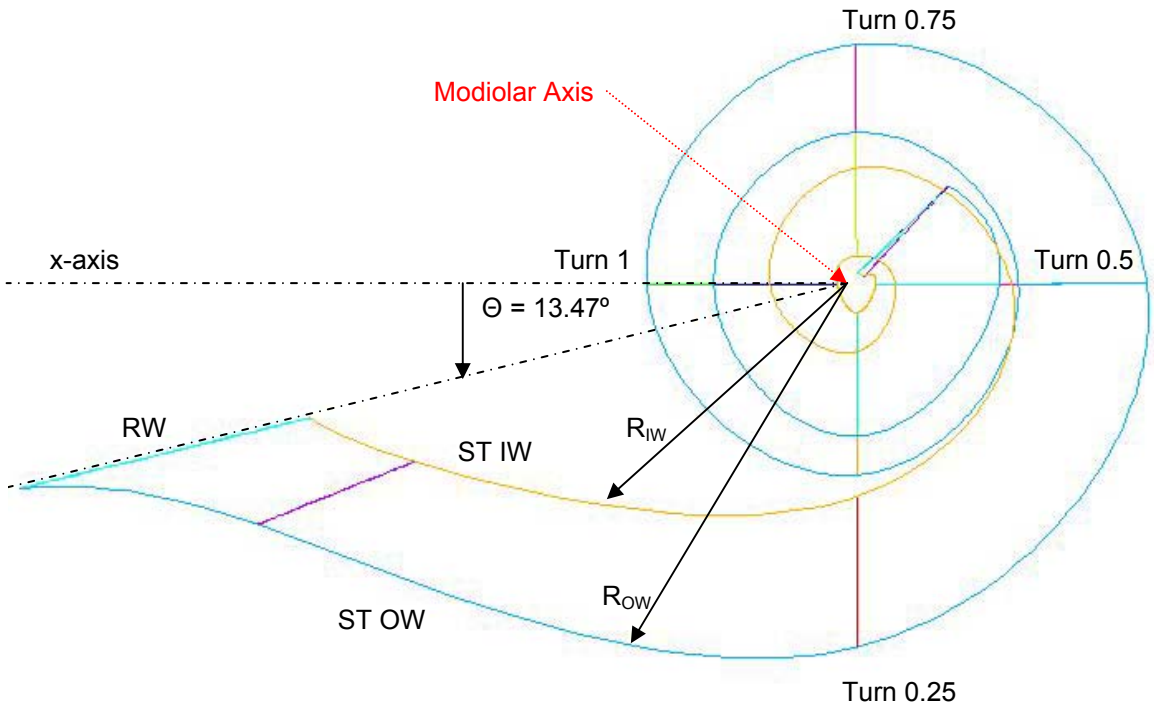

Figure 4.10. A 2D diagram of the ST (RW to turn 2.63; turns 0.25 to 1 marked), with measurements of ST radii, IW and OW lengths, percentage lengths and RW.

Table 4.1. IW and OW lengths [9], percentage lengths [9] and cochlear radii [11] for the length of the OC.

Turn	IW Length (L_{IW}) (mm)	IW Length (%)	OW Length (L_{OW}) (mm)	OW (%) Length	IW Radius (R_{IW}) (mm)	OW Radius (R_{OW}) (mm)
RW	0	0	0	0	5.66	
0.25	6.03	33.1	7.92	19.4	2.2	4
0.5	9.05	49.6	13.85	33.9	1.6	3.3
0.75	11.67	63.8	19.25	47.2	1.2	2.7
1	13.53	74	23.32	57.1	0.9	2.2
1.25	15.05	82.3	27.07	66.3	0.7	1.8
1.5	16.14	88.2	30.21	74	0.4	1.55
1.75	16.99	93	33.18	81.3	0.3	
2	17.67	96.7	35.81	87.8	0.2	
2.25	17.96	98.2	38.37	94	0.3	
2.5	18.17	99.4	40.25	98.6	0.2	
2.63	18.29	100	40.81	100		

The spiral starts at an angle of 13.47º at the site of the RW [11] (0º lies on the x-axis and angular displacement increases in an anti-clockwise direction from this point). The displacement from the 0º position at the tail of the angle vector (which represents the modiolar axis about which the cochlear spirals) is determined from the basal diameter of 7.96mm, which is obtained by averaging the mean values provided [3, 8]. This value is used to determine the starting distance along the 13.47º angle from the x-axis, as well as considering the radius versus insertion angle data provided by Cohen et al. [11] and is calculated as vector length 5.66mm at angle 13.47º. This starting position (coordinates of x = -5.54mm, y: height = 0mm and z = 1.33mm) and radii at dedicated angular positions along the cochlear spiral are originally determined by considering measured radii [11].

ST cross-sections are approximated and added to the model at dedicated angular distances about the modiolar axis. Initially, the cross-sections of the ST along its length are modelled as circles and then as ovals (with width greater than height for all measurements). However, neither approximation corresponded with measured results of

ST cross-sectional area [195]. Finally, the superior edge of the ST, considered to be the BM, is approximated as flat with insignificant thickness (for simplicity in modelling the ST) and the coordinates are mirrored to define the inferior edge of the ST. As the BM progresses towards the apex it increases linearly from 0.15mm (RW) to 0.52mm (apical end of OC or BM) [92]. The sides of the ST are modelled as parabolas. In the first approximation of the ST, it is assumed that the inner edge of the BM lies ¼ of the way along the width of the cross-section. The cross-sectional area is calculated at each quarter turn, by application of (11).

$$\text{Area}_{\text{parabola}} = 2\ H\ W/3 \tag{10}$$

$$\begin{aligned} \text{Area}_{\text{tot}} &= \text{Area}_{\text{rect}} + \text{Area}_{P1} + \text{Area}_{P2} \\ &= H\ W_{BM} + (W - W_{BM})\ H/6 + 3\ (W - W_{BM})\ H/6 \\ &= \mathbf{H\ W_{BM}/3 + 2\ W\ H/3} \end{aligned} \tag{11}$$

where H is ST cross-sectional height [195], W_{BM} is the width of the Basilar Membrane and W is the ST cross-sectional width [195]. For all cross-sections (from RW to 25mm along the ST length, at 0.5mm intervals), the approximations of the cross-sectional area determined using (11) compared well (the averages were close and within the upper and lower standard deviations) with measured results documented by Hatsushika et al. [195].

In the final model, the position of the BM is moved so that its inner edge is positioned at the centre of the cross-section, that is, at half the cross-section width. This corresponds to the centre of the ST, since Ketten et al. [8] define the centre of the cochlear duct to lie approximately where the BM and inner osseous lamina meet. Using this approximation, the coordinates of the cross-section change, as shown in the following diagram (Figure 4.11). However, the cross-sectional area remains the same (defined by 11):

$$\text{Area}_{\text{tot}} = \text{Area}_{\text{rect}} + \text{Area}_{P1} + \text{Area}_{P2}$$

$$= H\ W_{BM} + W\ H/3 + 2\ (W/2 - W_{BM})\ H/3$$

$$= \mathbf{H\ W_{BM}/3 + 2\ W\ H/3}$$

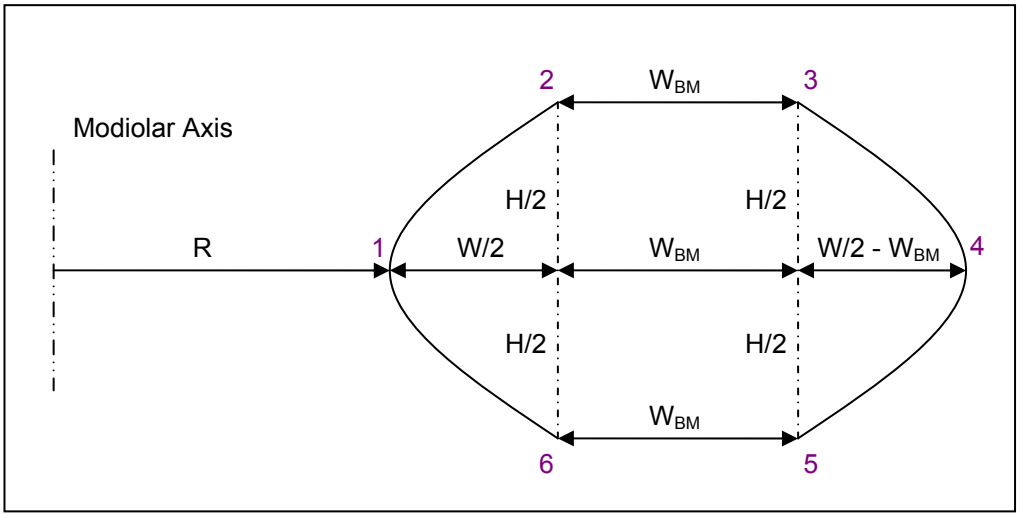

Figure 4.11. Cross-section of the ST, at radius R from the modiolar axis.

The cross-sections are defined at the RW and at each quarter turn, to the end of the OC (where the width and height data are extrapolated after ST length 25mm). Extrapolation is performed by analysing cross-section width and height information from the RW to 25mm along the OC and noting trends within OC length regions [195]. Analysis reveals that there is no significant change in cross-section width and height measurements after OC length 24.5mm. In effect, the values are kept constant after this length. A ST cross-section is added at turn 0.4 (radial position 14.4º), since ST height and width measurements decrease rapidly over the first 1.5mm of the ST length [195]. This produces more accurate results in the final model, where splines are used to interpolate between successive cross-sections and the dramatic changes in ST width and height in this region are captured. The model is extended to include change in axial height, which generally increases as the cochlea spirals upwards from base to apex.

Human cochleae vary in axial height from approximately 1.93mm to 4mm (mean 2.75mm) [8], as well as intra-cochlear tilt [7, 8]. Modelling cochlear height variation over radial distance is hence non-trivial. In this work, three variations are considered: a constant height increase to 2.75 mm, a constant angle of 10º and angle variation: from 0º to 360º (turn 0 to 1) there is a 0º incline, 360º to 780º (turn 1 to 2) a 10º incline and from 780º (turn 2 to 2.63) a 15º incline. Intra-cochlear tilt, similar to that illustrated in Figure 4.12, is introduced. A constant height inclination of 10º is combined with intra-cochlear tilt, to give a ST axial height of 2.56mm, which is close to the average of 2.75mm [8].

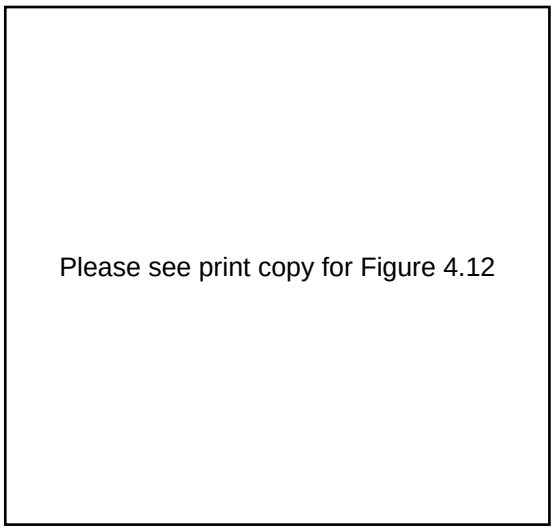

Figure 4.12. A cochlear cast showing intra-cochlear tilt [199].

In summary, previously published ST measurements [3, 8, 9, 11, 92, 195] are analysed for three-dimensional delineation of the chamber. A description of the ST is produced based on measured data, including ST cross-sectional height and width information, OC length, radial displacement about the modiolar axis, change in ST

height from base to apex and intra-cochlear tilt. Mean height and width measurements of the ST [195] are extrapolated to the average length of the OC, 35.58mm [9]. Results in [9, 11, 195] are combined to give cross-sectional measurements as a function of angular displacement about the modiolar axis. Height and width information is mapped to percentage lengths at each quarter turn, to the length of the OC (or 2.63 cochlear turn). The spiral begins at an angular displacement of 13.47º [11]: the location of the RW and common insertion site. At each defined cross-section, a value for radius (from modiolus to OC) is added by combining measurements of radii [11] with ST percentage lengths [9]. Each side of the cross-section is approximated by a parabola, with the BM surface kept flat and perpendicular to all height measurements. OC location is considered to be along the inner edge of the BM, where BM width variation is documented by Givelberg [92]. Intra-cochlear tilt and a constant 10º height inclination after the first turn are also included. This information gives a first approximation of the ST. The model is then parameterised to capture variations in ST size.

4.5.1.1 Parameterisation of the Model

A first approximation of the spiral is produced directly from measured data, as described previously. To make the model patient-specific and easily reproducible for differences in cochlear height, length and cross-sectional area, the dimensions of the cross-sections and their positions about the modiolar axis are made dependent on three parameters: ST length, width and height of the first cross-section (located at the site of the RW). Details for the process of model parameterisation are described in this section.

Kawano et al. [9] give mean percentage lengths of the OC and these values are used to determine OC length at each quarter turn, the RW and turn 0.04, taking the total length of the OC to be 35.58mm [9]. Average values for ST cross-section width and

height as measured by Hatsushika et al. [195] are used to derive percentage values based on the original cross-section, of width 3mm and height 2.42mm. The data [195] is extrapolated to the entire length of the OC. These percentages are then used to determine the width and height for each cross-section specified. For example, the cross-section at turn 0.25 (W_i) of average width 1.71mm and height 1.33mm gives the following percentage value (W%), which is used for parametric modelling of all cross-sections, based on only the width and height of the cross-section at the RW (W_{orig}):

$$W\% = (W_i / W_{orig})\ 100\ \% \quad (12)$$

$$= (1.71\ mm / 3\ mm)\ 100\ \%$$

$$= 57\ \%$$

Values for ST (OC) length and cross-section width and height are calculated using the above approaches and the values are documented in Table 4.2.

Table 4.2. Percentage lengths, cross-section width and height from RW to apex.

TURN	Width (%)	Width	Height (%)	Height	OC Length (%) [9]	OC Length from RW
0.00	100.00	3.00	100.00	2.42	0.00	0.00
0.04	57.00	1.71	55.00	1.33	4.20	1.49
0.25	50.50	1.52	48.00	1.16	21.80	7.76
0.50	45.50	1.37	42.50	1.03	36.30	12.92
0.75	41.50	1.25	38.00	0.92	49.10	17.47
1.00	41.50	1.25	39.50	0.96	58.80	20.92
1.25	41.50	1.25	34.00	0.82	67.30	23.95
1.50	41.50	1.25	32.50	0.79	74.50	26.51
1.75	41.50	1.25	32.50	0.79	81.10	28.86
2.00	41.50	1.25	32.50	0.79	87.30	31.06
2.25	41.50	1.25	32.50	0.79	92.60	32.95
2.50	41.50	1.25	32.50	0.79	97.30	34.62
2.63	41.50	1.25	32.50	0.79	100.00	35.58
2.75	41.50	1.25	32.50	0.79		

The values for cochlear radii, extending from the modiolar axis to the IW, at the RW, turn 0.04 and each quarter turn to turn 2.63, are originally plotted from measured data [11]. By analysing the measured radial displacements, values are calculated based

on the percentage length of the OC at the respective position. First, percentage values for radii (R%) are calculated based on average values of IW length (L_{IWi}) [9], measured radial values (R_i) [11] and percentage IW length [9]:

$$R\% \quad = R_i \; L\%_{IWi} / L_{IWi} \tag{13}$$

Radii (R_i) are then approximated based on ST length (L_{ST}), by application of (14):

$$R_i \quad = R\% \; L_{ST} / 2 \tag{14}$$

This yields the radii values tabulated in Table 4.3. Measured radii values [11] are provided for comparison with calculated results. Values that are used for calculation of ST cross-sections, including ST height coordinates (H / 2), Width coordinates (W / 2) and BM width (W_{BM}) are also tabulated. The latter assumes a linear increase along the length of the OC, from 0.15mm to 0.52mm [92].

Table 4.3. Values calculated for cochlear radii.

TURN	Radii (%) to IW (R%)	Measured Radii (mm)	Calculated Radii (mm)	H/2 (mm)	W/2 (mm)	W_{BM} (mm)
0.00	32.00%	5.7	5.69	1.21	1.50	0.1500
0.04	27.00%	5	4.80	0.67	0.86	0.1655
0.25	12.00%	2.2	2.13	0.58	0.76	0.2307
0.50	8.80%	1.6	1.57	0.51	0.68	0.2843
0.75	6.60%	1.2	1.17	0.46	0.62	0.3317
1.00	4.90%	0.9	0.87	0.48	0.62	0.3676
1.25	3.80%	0.7	0.68	0.41	0.62	0.3990
1.50	2.20%	0.4	0.39	0.39	0.62	0.4257
1.75	1.60%	0.3	0.28	0.39	0.62	0.4501
2.00	1.10%	0.2	0.20	0.39	0.62	0.4730
2.25	1.60%	0.3	0.28	0.39	0.62	0.4926
2.50	1.10%		0.20	0.39	0.62	0.5100
2.63	0.60%		0.11	0.39	0.62	0.5200
2.75	0.60%		0.11	0.39	0.62	

Table 4.4. Final coordinates defining the cross-sections of the three-dimensional, parametric model of the human ST.

		Coordinate:					Coordinate:		
turn	point	x	y	z	turn	point	x	y	z
0	1	-5.54	0.00	1.33	**1.5**	43	0.39	0.97	0.00
	2	-6.99	1.21	1.68		44	1.01	1.36	0.00
	3	-7.14	1.21	1.71		45	1.44	1.36	0.00
	4	-8.45	0.00	2.03		46	1.64	0.97	0.00
	5	-7.14	-1.21	1.71		47	1.44	0.58	0.00
	6	-6.99	-1.21	1.68		48	1.01	0.58	0.00
0.04	7	-4.47	0.00	1.77	**1.75**	49	0.00	1.77	-0.28
	8	-5.26	0.67	2.08		50	0.00	2.17	-0.91
	9	-5.42	0.67	2.14		51	0.00	2.17	-1.36
	10	-6.06	0.00	2.40		52	0.00	1.77	-1.53
	11	-5.42	-0.67	2.14		53	0.00	1.38	-1.36
	12	-5.26	-0.67	2.08		54	0.00	1.38	-0.91
0.25	13	0.00	0.00	2.13	**2**	55	-0.20	2.53	0.00
	14	0.00	0.58	2.89		56	-0.82	2.92	0.00
	15	0.00	0.58	3.12		57	-1.29	2.92	0.00
	16	0.00	0.00	3.65		58	-1.44	2.53	0.00
	17	0.00	-0.58	3.12		59	-1.29	2.13	0.00
	18	0.00	-0.58	2.89		60	-0.82	2.13	0.00
0.5	19	1.57	0.00	0.00	**2.25**	61	0.00	2.53	0.28
	20	2.25	0.51	0.00		62	0.00	2.92	0.91
	21	2.53	0.51	0.00		63	0.00	2.92	1.40
	22	2.93	0.00	0.00		64	0.00	2.53	1.53
	23	2.53	-0.51	0.00		65	0.00	2.13	1.40
	24	2.25	-0.51	0.00		66	0.00	2.13	0.91
0.75	25	0.00	0.00	-1.17	**2.5**	67	0.20	2.53	0.00
	26	0.00	0.46	-1.80		68	0.82	2.92	0.00
	27	0.00	0.46	-2.13		69	1.33	2.92	0.00
	28	0.00	0.00	-2.42		70	1.44	2.53	0.00
	29	0.00	-0.46	-2.13		71	1.33	2.13	0.00
	30	0.00	-0.46	-1.80		72	0.82	2.13	0.00
1	31	-0.87	0.00	0.00	**2.63**	73	0.07	2.93	-0.08
	32	-1.49	0.48	0.00		74	0.50	3.33	-0.53
	33	-1.86	0.48	0.00		75	0.86	3.33	-0.91
	34	-2.12	0.00	0.00		76	0.93	2.93	-0.99
	35	-1.86	-0.48	0.00		77	0.86	2.54	-0.91
	36	-1.49	-0.48	0.00		78	0.50	2.54	-0.53
1.25	37	0.00	0.53	0.68	**2.75**	123	0.00	3.33	-0.11
	38	0.00	0.94	1.30					
	39	0.00	0.94	1.70					
	40	0.00	0.53	1.92					
	41	0.00	0.11	1.70					
	42	0.00	0.11	1.30					

The cochlear axial height, including a 10º constant height increase after the first turn and intra-cochlear tilt, is re-calculated using ST length calculations from Table 4.2. The new cochlear axial height is 2.93mm, which remains close to the average (2.75mm [8]). This height information is used to plot the cross-sections about the modiolar axis. ST cross-section coordinates are re-calculated considering the new values for cochlear radii (Table 4.3), cross-section width and height (Table 4.2), as well as variation in axial height. The position of the ST cross-section coordinates relative to the cochlear radii are defined in Figure 4.11. The helicotrema is represented as a single point at turn 2.75. Results for the positions of the ST cross-section coordinates are tabulated in Table 4.4.

In review, the parametric model of the ST is derived by calculating cross-section width and height values as percentages of the initial cross-section, at the location of the RW. Radii values, extending from the modiolus to the ST IW, are calculated as percentages of OC length. Therefore, to derive a patient-specific model of the ST using the method described previously, three parameters are required:

1. OC length, defined as the centre of the fluid-filled cochlear chamber
2. ST cross-section width at the RW site, measured perpendicular to the BM
3. ST cross-section height at the RW site, measured perpendicular to the BM

These parameters may be measured from pre-operative CT scans of the CI candidate.

A number of assumptions are made in generating the model:

1. Radii percentage lengths are as detailed in Table 4.3 and do not vary
2. Radii lengths are calculated based on OC length
3. Height, if not otherwise defined, is assumed to increase according to a constant 10º with intra-cochlear tilt

4. Cross-section widths and heights along the ST length are defined as a set percentage (Table 4.2) of the original cross-section width and height at the RW
5. A linear increase in BM width, as it spans the length of the OC from base to its most apical point (turn 2.63). It is also assumed that the BM exists for the entire length of the OC
6. BM initial and final width are defined by measured data, unless otherwise specified [92]

The parametric model produced closely approximates the spiral generated by plotting the measured data [3, 8, 9, 11, 92, 195], with respect to ST cross-section width and height, axial height, radii and ST length.

4.5.2 FEM of the ST in ANSYS

A surface description of the parametric ST model is generated in ANSYS. ANSYS is selected for model construction as it provides facilities to fully develop a three-dimensional surface structure and enables the export of models in VRML format. Haptic-rendering is to be performed in the Reachin API, which supports VRML scripting for scene-graph development. In effect, ANSYS is an effective tool for accurate model generation and cross-platform support.

For model creation in ANSYS, the coordinates for each ST cross-section are imported into ANSYS as *keypoints*. *Splines* are then drawn between all keypoints that define one cross-section, starting at point 2 and progressing anti-clockwise through points 1, 6, 5, 4, 3 (refer Figure 4.11 for point location). Points 2 and 3 remain unconnected as this is the site of the BM and assumed to be membranous. The BM is modelled as a separate structure, by defining the area along the top of the ST spiral.

Splines are used to connect each ST cross-section, from base to apex. The ANSYS function *LOVLAP* is applied to segment all the splines at the intersecting keypoints, for creation of the ST surface. Surfaces are formed between the spline segments by choosing the option *create Areas* and selecting the four bordering splines within which the surface area is to be defined. This is done for the entire length of the ST. A macro may be executed to speed up the process of keypoint, spline and area definition. A three-dimensional surface description is the final product, as shown in Figure 4.13. For the purpose of visual rendering (to be discussed in the next chapter), a course mesh is created over the surface area (Figure 4.14), by defining an element type SHELL93 and using the Mesh Tool, with options *Quad* shape and *smart size* set.

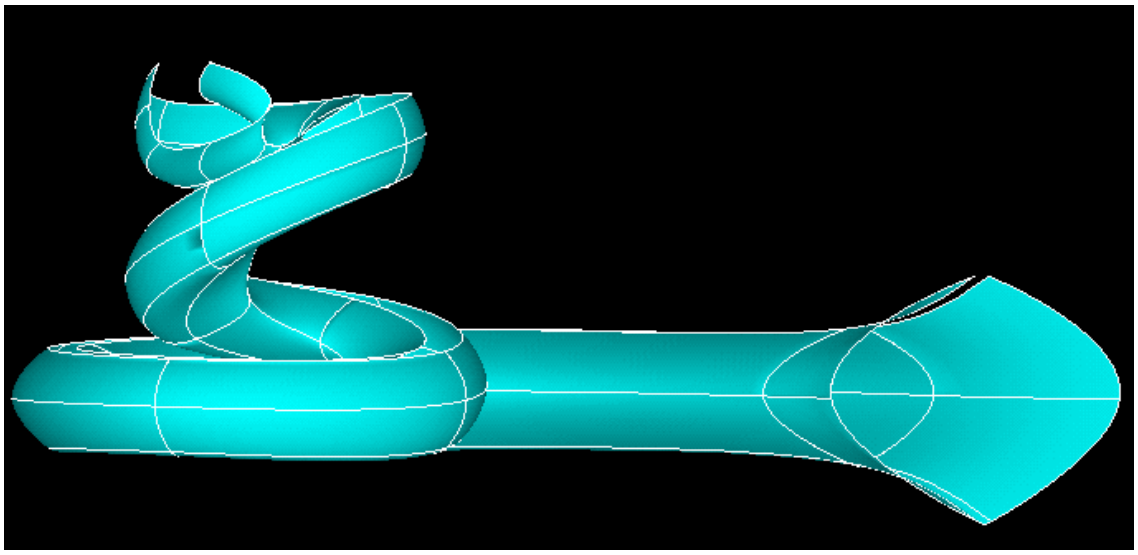

Figure 4.13. Three-dimensional ST surface reconstruction produced in ANSYS.

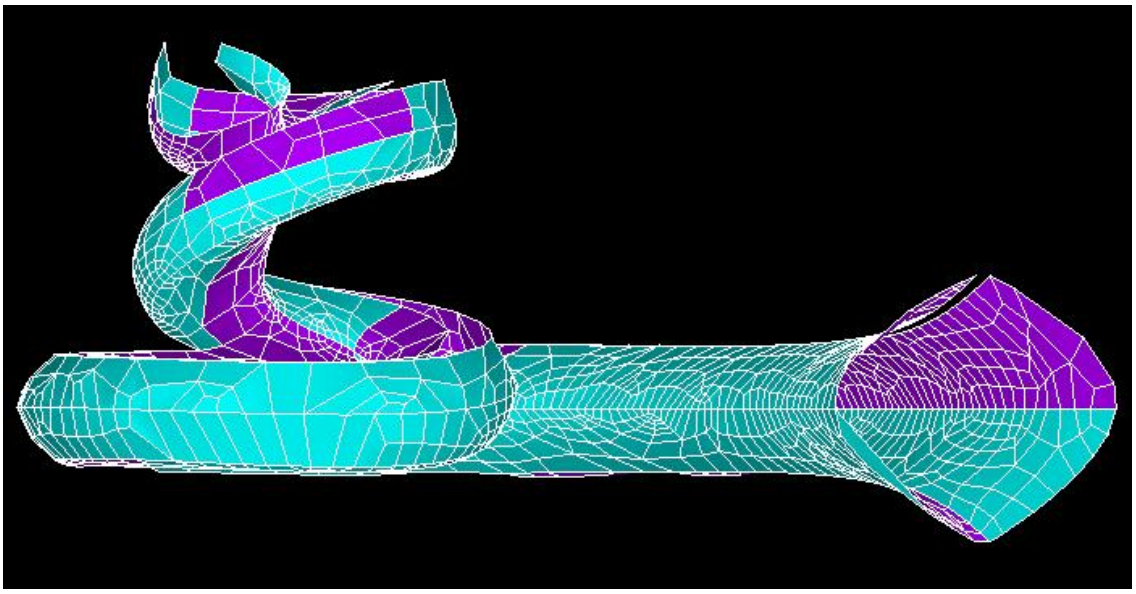

Figure 4.14. Finite Element Mesh produced from ST surface area in ANSYS.

The accuracy of the ST model has been revealed earlier, where cross-section areas, spline lengths and cochlear radii were compared with measured results. The design is also compared with two mathematical models discussed previously: the helico-spiral approximation [7] and the Archimedian model [8]. These three-dimensional spirals define the central path of the cochlea, which would lie within the region of the CP. The parametric data produced for representation of the ST cross-sections are plotted along the length of the spirals, with the top of the cross-sections aligned with the spiral central path estimation. The information is replicated in ANSYS and surface descriptions of the helico-spiral and the Archimedian spiral are produced in Figures 4.15 and 4.16 respectively. Cross-section data from this work plotted along the helico-spiral path [7] gave a favourable three-dimensional representation of the ST, which is similar to the model produced in this work (Figure 4.13). There are noticeable differences between the representations, however, near the base (since the base of the

cochlea flattens out and is not truly logarithmic in this region [7]) and the inclusion of intra-cochlear tilt in this work, which is not considered by Yoo et al. [7] or Ketten et al. [8]. The same cross-sections are interpolated along the Archimedian spiral [8] to give the result in Figure 4.16. In this case, the model suffered inaccuracies near the helicotrema and so the cross-section data are only plotted to turn 2.25. The spiral [8] actually traces an electrode array trajectory as described by Cohen et al. [11], which differs from the cochlear central path. This may explain the inconsistencies between results in this work and that of Ketten et al. [8].

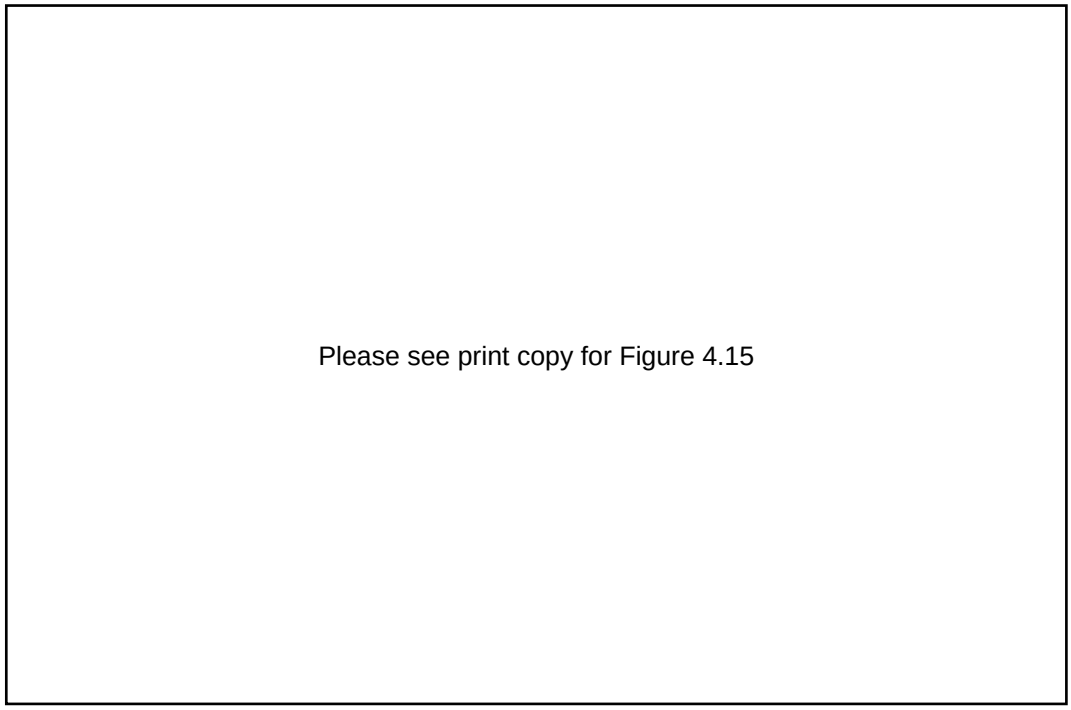

Figure 4.15. Cross-section data plotted in ANSYS along the Helico-spiral Approximation [7] to produce a surface reconstruction of the ST.

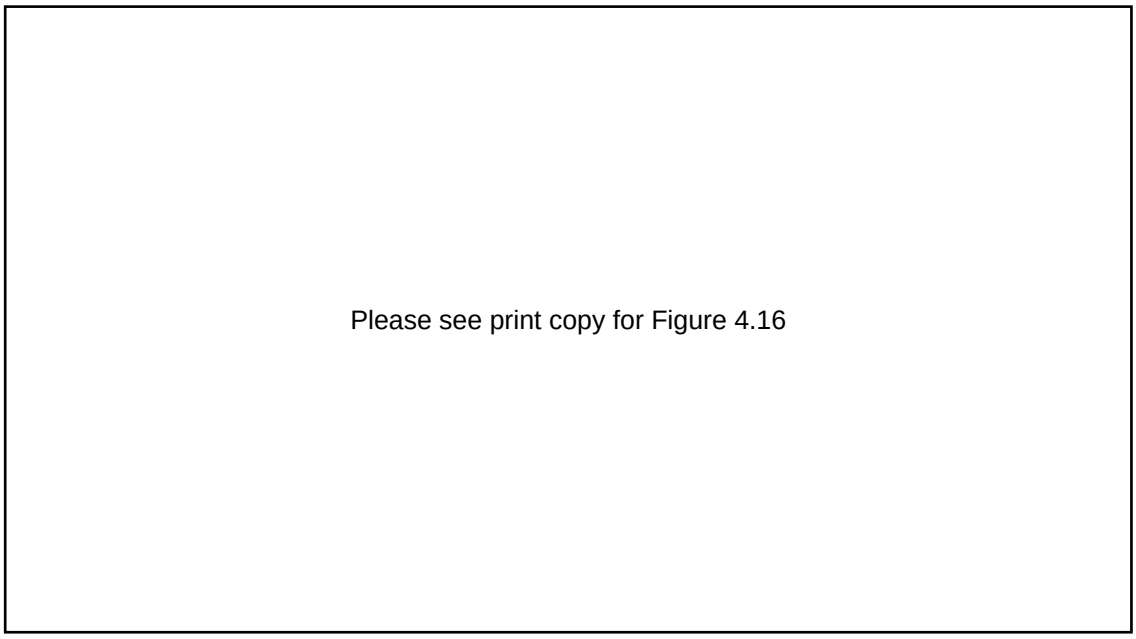

Figure 4.16. Cross-section data plotted in ANSYS along the Archimedian spiral [8] to produce a surface reconstruction of the ST.

A three-dimensional description of the silicone carrier which houses the electrode array is also produced in ANSYS (Figure 4.17) for the virtual CI insertion, using a similar process to define keypoints, lines/splines and areas between intersecting lines/splines. The dimensions of the array are taken from the documentation [25, 31], as well as provided by Cochlear^{TM}. The Nucleus® 24 Contour electrode design tapers from a tip diameter of 0.5mm to 0.8mm at the end of the embedded electrode array [25, 31]. The 22 individual electrodes are placed along the first 15.5mm length of the carrier [31]. Each electrode has a width of 0.3mm [31], a geometric area between 0.28mm^2 and 0.31mm^2 [31] and the inter-electrode spacing varies from 0.1mm at the tip to 0.48mm at the proximal end of the array [25]. The three silastic marker ribs are at a proximal position to the electrodes: the first rib is at a distance of approximately 22mm from the tip (approximately 1mm to 2mm more when it is curved/relaxed) (Cochlear^{TM}, 2004).

The distance between the silastic bands is 0.5mm. The bands are 0.5mm in thickness and the diameter of the bands is 1.1mm (Cochlear^{TM}, 2004). The diameter of the silastic cavity that contains the stylet ('lumen') is 0.15mm and its length is close to 31mm when the electrode is straight (approximately 1mm to 2mm more when it is curved or relaxed). For the purpose of virtual CI insertion in this application, the entire length of the carrier is 45mm (in reality it is longer however only the region of the electrode array, marker ribs and stylet is required for the simulation). This information is used to model the Nucleus® 24 Contour geometry in ANSYS.

The stylet is modelled as a separate structure. It is 0.18mm in diameter and its length from the tip of the stylet to the start of the loop is nominal 32mm (Cochlear^{TM}, 2004). The inside diameter of the stylet loop, which is located for the surgeon to grasp at the proximal end of the stylet, is between 0.63mm to 0.75mm (Cochlear^{TM}, 2004). Surface reconstructions of the carrier and stylet produced in ANSYS are shown in Figure 4.17. A surface representation of surgical tweezers has also been constructed in ANSYS for use in the simulation, for the user to 'grasp' the end of the silicone carrier. A pair of tweezers was physically measured and the approach for its reconstruction in ANSYS is similar to that described for the electrode carrier and stylet.

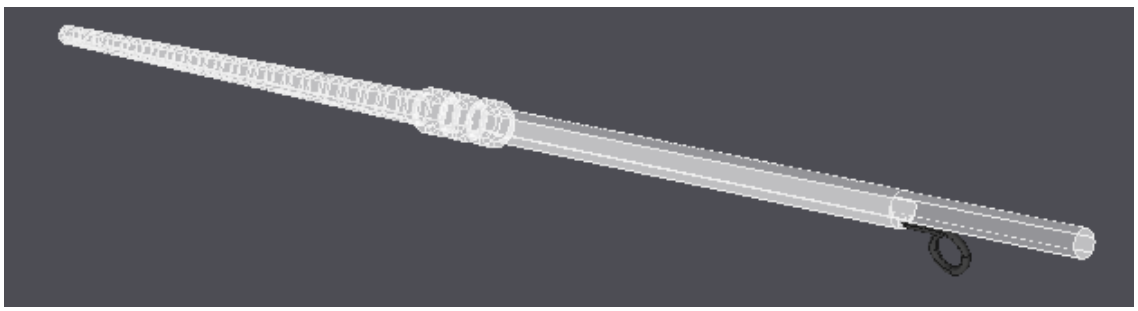

Figure 4.17. A surface description of the Nucleus® 24 Contour^{TM} with stylet, created in ANSYS.

The design and construction of a virtual three-dimensional surface representation of the ST has been discussed in this section of work and compared with existing data. The surface description of the ST model is sufficient for this application, since its geometry will not be changing form during the insertion; only the carrier will deform during contact with the ST walls. A surface-based description as opposed to a volume-based model will reduce computation time, which is of particular importance for real-time haptic rendering. The parametric ST model is reproducible and requires three values for its creation: OC length, ST cross-section width and height at the site of the RW. A final model of the ST structure produced in ANSYS, as well as surface descriptions of the BM, electrode array carrier and stylet, are exported as VRML scripts (WRL file format) for modification, visual rendering and force modelling of the CI insertion in the Reachin API. Visual and haptic rendering of this medical procedure are examined in the next chapter.

Chapter 5

VISUAL AND HAPTIC RENDERING OF THE COCHLEAR IMPLANTATION

5.1 Introduction

Once the geometric model of the human ST is created, the next stage in system design is to make the scene more dynamic by enabling real-time interactions between the user and model in a virtual environment. This process is comprised of two important, distinctive features: visual rendering and haptic rendering. Both of these phases must be introduced to provide the user with a realistic, immersive virtual environment that pertains to both visual and physical properties that replicate real-world characteristics. In this section of work, the development of a surgical simulator for training specialists in cochlear implantation is discussed, including system design, ST model optimisations and program features for enabling interactive, real-time insertion of a virtual implant into the model, with visual and force feedback delivered to the user during electrode advancement.

To synthesise visual and haptic rendering of CI surgery using the geometric model of the ST, a supporting infrastructure including dedicated software and hardware is required. The set-up of the surgical simulator consists of specialised software that facilitates the visual and haptic rendering of the cochlear implantation: the Reachin Application Programming Interface (API). It requires specific hardware to run which

has been installed for this purpose, including a 6DOF haptic device. Stereoscopic viewing further enhances the user's experience.

5.1.1 Reachin API

Reachin API is a software package that facilitates fast and realistic three-dimensional modelling of virtual reality applications. It is selected as the design platform for simulator construction, including visual and haptic rendering of cochlear implantation, as it offers a number of benefits for this type of application. Reachin API uses the concept of a single scene-graph to describe a real-world environment that has both graphical and physical attributes. Using this design, visual and force modelling may be implemented in one software environment. Graphics and haptics are modelled in the same scene-graph and share the same data, however the two rendering engines are kept separate. A scene-graph has a hierarchical data structure and is composed of *nodes* and *fields* to describe the scene it represents. *Nodes* are scene graph primitives that store information in *fields*. Relationships between fields are established via field networks. These enable the realisation of data dependencies and information flow such as event propagation or field initialisation. The scene-graph concept can speed up program development time and also offer optimisations so that rendering performance is enhanced.

Real-world characteristics may be attributed to objects defined within the virtual scene. Object shape properties including geometry, appearance (such as colouration) and position may be defined, as well as features of the scene itself (lighting effects, for example). Dynamic properties can also be modelled (such as inertia and magnetism), as well as surface attributes (including roughness and stiffness). Reachin API offers a number of custom nodes (specifically 6DOF force models) for fast and simple

development of different geometries and associated behaviour. Users can modify and create nodes for specialised models, deriving properties from existing nodes as well as introducing new features. For model implementation, programming is done using C++ which is integrated with two scripting languages: VRML and Python. VRML enables a simple description of the scene to be rendered, whilst python scripting is mainly used for field event handling. Borland C++ Builder 6 (Borland Software Corporation) is installed for software development on the Windows XP operating system. Reachin API requires Borland C++ Builder 5/6 and does not currently support Microsoft Visual C++.

Reachin API uses two rendering engines: the haptic rendering loop referred to as the *real-time loop* and the graphics or *scene-graph* loop. Collision traversal and graphics rendering takes place in the scene-graph loop while force calculation and delivery is performed in the real-time loop. Traversal of the real-time loop occurs at a frequency of 1000Hz whilst the scene graph loop is updated at 30Hz [104]. Synchronisation of the graphic and haptic loops is done by the Reachin API so the user does not need to perform this task. Using an OpenGL/Ghost solution, manual synchronisation would be required and separate datasets exist for the two loops. In the Reachin API, although these two loops are kept separate, they share the same dataset. This avoids issues concerned with artifacts and increases computer processing speeds, whilst sparing the user valuable development time in avoiding synchronisation management. Reachin API offers the developer numerous benefits that facilitate fast and realistic model generation in virtual reality applications.

The structure of the Reachin API lends itself to visualisation and physical representation of objects using surface-rendering solutions as opposed to volume-based approaches. This is due to the hierarchical data arrangement of node and field constructs with direct utilisation of VRML and python scripting. The portability of VRML is also a

desirable feature for model construction. The advantages and disadvantages of both surface-based and volume-based approaches have been discussed in Chapter 2. A fundamental factor for deciding whether to use surface or volume representations of the data is the application. For this work, the process of CI insertion may be viably implemented using a surface rendering approach. The ST passage does not transform during an insertion (such as significant soft tissue deformations [100, 102, 132, 137, 139, 140, 149, 150, 156, 158, 164-169, 171-173, 178, 180, 182-185]) and can be represented by primitives that have physical attributes. The Contour silicone carrier can bend, flex and transform during an insertion. Despite this observation, assuming that the inside of the carrier volume remains unchanged, variation in surface topology of the carrier as it is inserted into the ST can be modelled using surface rendering techniques with polygonal primitives.

5.1.2 Hardware

To support the Reachin API, the system runs on a Deltacom DSS1100 with dual 2.4GHz Intel Xeon processors, 800MB Random Access Memory (RAM), 80GB of hard disk space and a Quadro4 700XGL graphics card. A 6DOF Phantom Premium haptics device provides force and torque feedback to the user. An in-house framework supporting a monitor and mirror enables co-location of the haptics arm and virtual stylus. A mirror function is provided in the Reachin API to provide correct reflection of the image that is viewed by the user (so that it is not an upside-down display). A monitor splitter has been included for ease of system development. Stereovision is supported using CrystalEyes shutter glasses with emitter (Direct2U) for three-dimensional stereoscopic viewing. This is included to provide the user with a more realistic scenario of the operation. The surgeon would normally look through a

microscope during the stages of a cochlear implantation. Stereoscopic vision is supported by the Reachin API and is introduced to further enhance the user's sense of immersion into the virtual environment. The hardware set-up is shown in Figure 5.1.

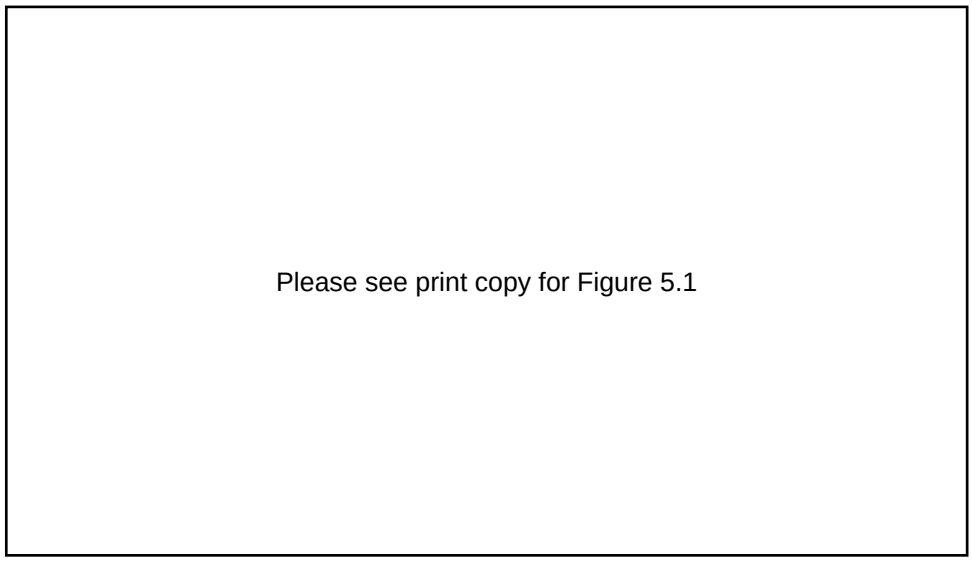

Figure 5.1. The system hardware for the simulator, including a 6DOF haptics device, custom framework for tool and stylus co-location, and stereo viewing.

5.2 Simulator Realisation

5.2.1 Preliminary Visualisation and Interaction

Surface models of the ST, BM, Nucleus® 24 Contour electrode and stylet, produced using ANSYS were exported as VRML 1.0 files and converted to VRML 2.0 using the Windows program *vrml1tovrml2*, by execution of the command line:

>> vrml1tovrml2.exe *infile.wrl outfile.wrl*

Reachin API does not support VRML version 1.0. The script files were modified and visually rendered in the Reachin API. Alterations to the VRML version 2.0 files were

made to run the program in the Reachin API, as well as reduce file size (hence processing time) and allow variations in object colour. These changes included manual removal of *Color* and *Normal* nodes, along with respective index fields. Nested *Transform* nodes were added, to enable scaling, translation and rotation of the different structures. Colouration was varied in the *diffuseColor* field of the *Material* node.

To facilitate initial interactions with the model, *SimpleSurface* nodes were added to the ST surface fields, providing haptic feedback to the user when the device tool tip was in contact with the model. Surface elasticity was attributed to the BM by adopting the *MembraneIFS* node within the *geometry* field of the *Shape* node. The *stiffnessM* field of the *MembraneIFS* node, which defines the extent of surface deformation about the point of contact [104], was set to a value of 100. The initial ST and BM representations in the Reachin API are shown in Figure 5.2. Visualisation of the Contour array and stylet was performed using the same process (file conversion, modification and program execution). All structures were first rendered separately and then combined into the one VRML scene-graph for display and interaction, as shown in Figures 5.3 and 5.4.

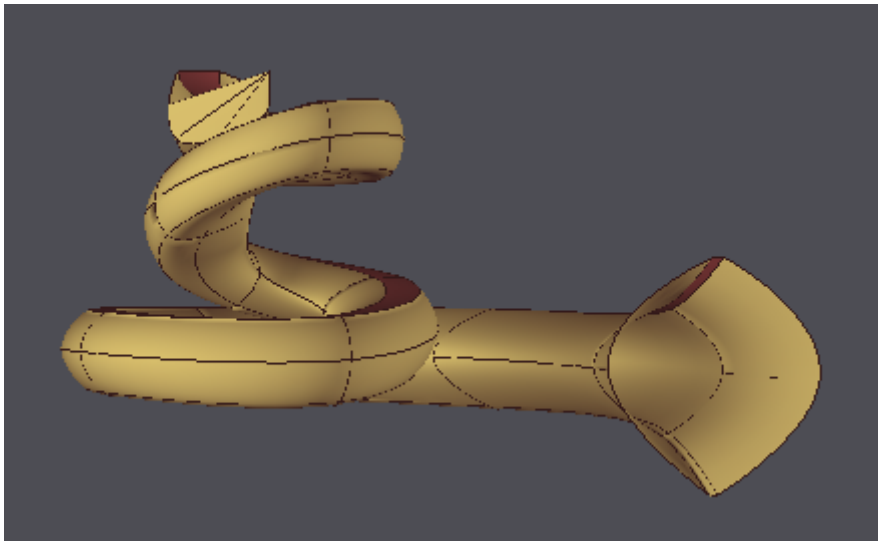

Figure 5.2. Model of the ST and BM structures shown in Reachin API scene (note the additional surfaces at the apex, representing the helicotrema).

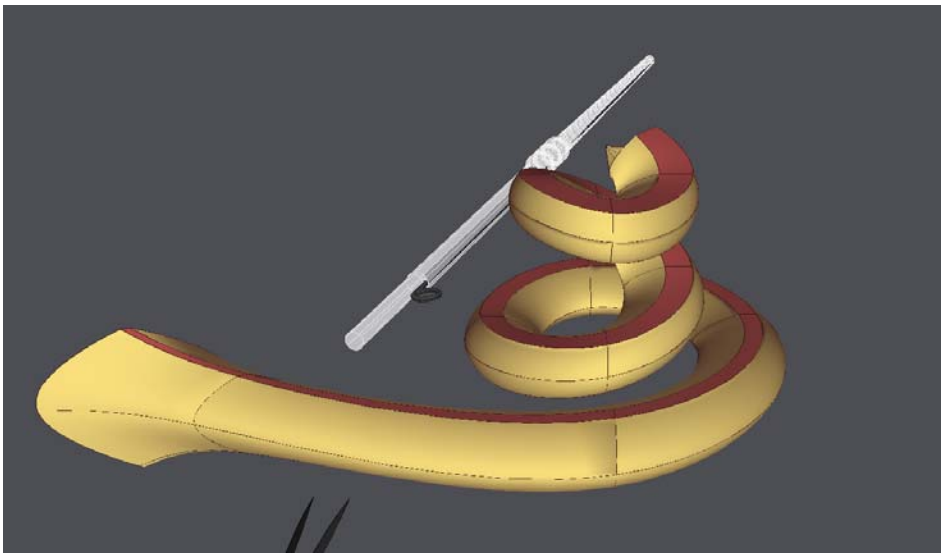

Figure 5.3. Simulator GUI in the Reachin API with the ST, BM, electrode carrier and stylet. A top and front view of the scene.

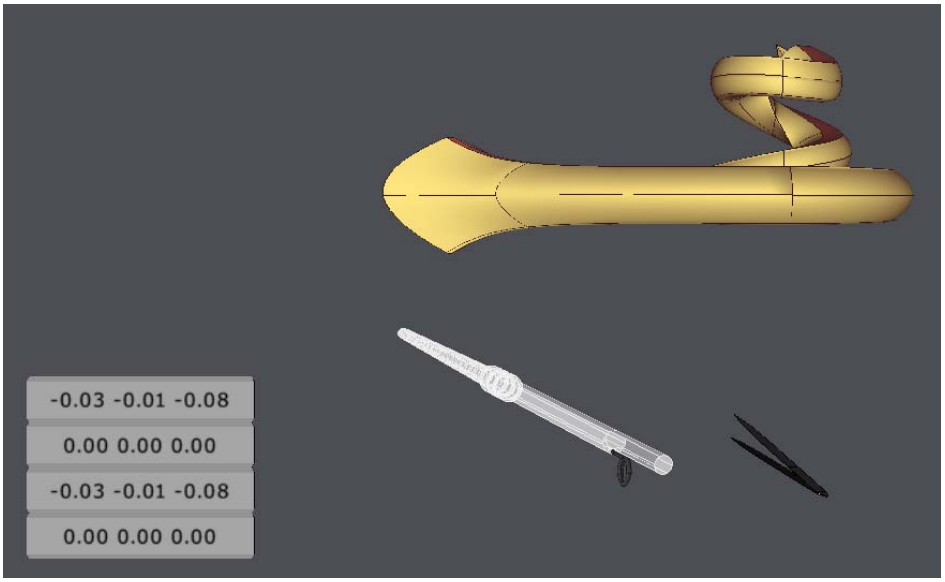

Figure 5.4. Simulator GUI with the ST, BM, electrode carrier and stylet, as well as tweezers and force/position display. A front view of the scene.

Python script was written to enable real-time rotation and translation of the model via the keyboard during program execution (alternatively, a space mouse may be used). Keys 'i', 'j', 'k' and 'l' were used for object rotation up, left, down and right respectively. The key 'u' translates the object to the left and 'p' to the right. Rotation and translation are about the x- and y- axes. This functionality was implemented by accessing the *rotation* and *translation* fields of the objects' *Transform* node (Appendix C, 'RotateCochlea.py').

5.2.2 Dynamic Environment

Following preliminary object visualisation, the second phase in simulator development was to make the simulation more dynamic. This involved detecting object collisions within the scene and generating appropriate visual and force responses during advanced object interactions. This was done for stylet removal as well as array insertion into the

ST. For ease of code reproduction, modification and extensibility, customised nodes were developed in C++ (Appendix D) for the ST (files STympV.h, STympV.cpp, STympH.h, STympH.cpp), BM (files BMembV.h, BMembV.cpp), Contour array (files CArr.h, CArr.cpp) and stylet structures. These nodes were combined under one primary node, *CochlearV* (Appendix D, files CochlearV.h, CochlearV.cpp) and used in a single VRML file to provide a brief description of the scene (Appendix E, file run_simulation10.wrl). A series of fields were exposed in C++ for access in VRML and python script, including the *scale* and *translation* fields of each structure. The programs were compiled using the Makefile (Appendix F, file Makefile), which is a modified version of the Makefile provided by Reachin Technologies [104]. A program executable was generated for scene display, using the main program CochlearBuilder.cpp (Appendix G) for scene generation [104]. The executable was run from the command prompt via:

```
>>      cochlear_node.exe run_simulation10.wrl
```

The file 'run_simulation10.wrl' contains function calls that have not yet been described. These functions will be discussed in the following sections.

5.2.3 Contour and Stylet Mobility

Change in orientation of the cochlea (comprising ST and BM) is slight during the actual surgical procedure. The surgeon may tilt the patient's head or alter viewing angle during a CI insertion, yet this is not to a significant degree. Cochlear movement via keyboard input, implemented in python script (described previously), is therefore sufficient for this application. Contour array and stylet mobility are, however, not slight but vary arbitrarily as they are moved about the scene. The user will dictate the movement of the array and stylet in real-time, during program execution.

User-induced movement of the array and stylet was first implemented using the *GraspableTransform* node, within the *Contour* and *Stylet* node definitions. During run-time, the user enables 'attachment' of the two structures to the tool tip by pressing the haptic device button. This functionality was implemented by making the *Transform* nodes of the structures *GraspableTransform* nodes in the object surface descriptions (wrl files). The *GraspableTransform* node in the object class was also defined, within the header file and the fields for *global_grasp_position* and *global_grasp_orientation* were exposed in the Interface declaration within the header file. Field routing was established between the haptic device position (*trackerPosition*) and orientation (*trackerOrientation*) with the *GraspableTransform* position and orientation fields. The *lazy evaluation* of the field routing meant that when the device button was pressed, routing was enabled and when it was released, fields were unrouted. For the user, holding the button down whilst moving the device allowed for movement of the array and stylet about the GUI, while button release disables structure motion. The *GraspableTransform* node, however, did not allow for accurate movement and positioning of the structures.

To replace the *GraspableTransform* node functionality, Python script was written to more accurately implement Contour and stylet mobility, by accessing the *Display* node *trackerPosition* and *trackerOrientation* fields independently of the *GraspableTransform* node. These fields of the *Display* node were routed to the *Transform translation* and *rotation* fields of both objects whilst the button was pressed and field unrouting performed for button release. This enabled positioning of the array and (offset) stylet at the tip of the avatar and along the longitudinal axis of the device during array selection. The array and stylet could then be moved precisely about the screen, with correct orientation and final placement. The original file which was written

to implement this functionality is contained in Appendix H (file 'MoveCarrierR4AB.py'). The Stylet and Carrier nodes are passed to the function as reference '0', with separate function calls to the python file for each structure. A modified version of the program is given in Appendix E (file 'MoveCarrier_noCA11.py'). In the latter program, the carrier is optimised by a sub-sampled version of the array, which is discussed in section 5.2.7 Program Optimisations. The latter program is used in the final version of the simulation. The stylet structure is omitted (commented in the file) and the reasons for this will be given in section 5.2.7.

5.2.4 Surgical Tools

Live electrode array insertion is performed using either jeweller's forceps or an inserting claw. A right-angled hook is used for stylet withdrawal. ANSYS was again employed to construct forceps (two structures for closed and open states) and a hook in a process similar to that described in Chapter 4, section 4.5.2. Again, these files were converted to VRML 2.0 format, slightly modified and visualised in the Reachin API. The VRML files were imported into the *Cochlear* VRML scene-graph, using the *Import* node to facilitate fast program editing and script brevity (Appendix E, file 'run_simulation10.wrl'). Tool selection is performed during run-time via the keyboard and is implemented in VRML and Python (Appendix E, files 'run_simulation10.wrl' and 'MoveCarrier_noCA11.py', which are the current versions of the files, used in the simulation). A *Switch* node is employed in the *Cochlear* scene-graph for stylus and tip representations. Upon user selection, the *whichChoice* fields of each are set to the respective tool in the python script. Similar code is added to the Python script (Appendix E, file 'MoveCarrier_noCA11.py') that monitors the device button state,

with forceps in (clasping the end of the array) when the button is pressed and out upon release. The reference to the *Switch* node is termed *SwitchStylus* in 'MoveCarrier_noCA11.py'. The tweezers are shown in Figure 5.4, with its tips visible in Figure 5.3.

5.2.5 Stylet Withdrawal

During advancement of the Contour array into the ST using the SIT, the stylet is removed after full electrode insertion. Stylet withdrawal is achieved in practice by hooking the stylet loop and pulling it gently out of the carrier. Prior to removal, the stylet holds the silastic carrier straight. During stylet withdrawal, the Nucleus® 24 Contour™ coils to its pre-curled state, towards the exposed electrode array orientation. Contour curling is designed to position and orient the array towards the modiolus. This is done to reduce cross-channel interference and electrode current levels [25]. Restoration forces are projected during silastic recoil and help secure its final position.

The process of stylet withdrawal presents the challenges of object collision detection and response. The primary challenges of this procedure include: determination of stylet (loop) location and detecting tool tip/object interactions including the linear movement of the stylet during its withdrawal. Reachin API provides custom nodes that are useful for this type of problem. As a first approach, the *CollidingController* node was used for the implementation of collision detection and response during interactions between a right-angled hook and the stylet. The *CollidingController* node is provided in the Reachin API node library and enables the virtual tool to be described by different geometries (other than a single sphere). Its primary function is to detect and react to collisions during object interactions. The tool geometry is specified within the *points* field of the *CollidingController* node and the object/s with which the tool physically

interacts is/are defined in the *CollisionChild* node, within the *CollidingController*. This means that the points field must be updated during any object transformations (such as stylet withdrawal along a straight line projection) or tool change.

For the process of detecting the action of hooking the stylet loop, the use of the *CollidingController* node did not work successfully for the large number of points defined in the *Stylet* VRML representation. Interaction with the device often proved unstable, leading to forces in excess of 8.5N for an object of small scale (less than 0.02m) and complex geometry. This resulted in program termination (including generation of a 'RUNAWAY(1)' error). Throughout program execution, tool motion was not smooth. The haptic representation of the stylet was simplified as a single loop, constructed as an *IndexedLineSet* (using ANSYS). This was done to reduce the number of polygons and hence the time taken to traverse the data.

Stylet withdrawal was first implemented using the *ButtonSimpleSurface* node to move the stylet as it is touched with the tool tip. The *ButtonSimpleSurface* node is provided in the Reachin API library and its properties are derived from the Reachin API node *ButtonSurface* with a *SimpleSurface* base class. Its *armed* field is set when the button surface is pressed and the *activate* field is set upon button release. Python script was written to access, route and set custom fields exposed in the *Stylet* node with real-time values. This facilitated stylet withdrawal, however the loop surface had to be 'hit', which is not good practice, nor does it mimic real-life stylet removal. Using this surface description, the *CollidingController* node gave good visual hook representation, yet the points of contact for haptic interactions were not precise.

The implementation for hook and stylet interactions using the *CollidingController* node with the *ButtonSimpleSurface* node is provided in Appendix I. Within the file 'Cochlear_colcont.wrl' the *CollidingController* node is defined, with an

IndexedLineSet geometry and the *points* field set to the same coordinates. The *CollisionChild* is defined as *LOOP_TRANS* which is a simplified *IndexedLineSet* haptic representation of the stylet (refer to the file 'centre_loop_2.wrl'), with a *ButtonSimpleSurface* surface for haptic interactions. The file 'LoopTouchedR4.py' implements the functionality to deal with the button surface being armed and activated during hook interactions. This includes changing the *translation* field of the stylet whilst the button surface of the stylet loop is armed. Once the armed field is set, the visual representation of the stylet and haptic representation of the loop are translated along a linear trajectory, away from the silicone tip of the array, mimicking gradual stylet withdrawal. Field routing for the button pressing is established at the end of the file 'Cochlear_colcont.wrl'. Selection of the hook is performed via the keyboard and is implemented in the switch statement in 'Cochlear_colcont.wrl', with a python call to the file 'SelectToolR4.py'. The program determines which tool to show at the GUI based on the key that is selected by the user.

Finally, smooth and gradual stylet withdrawal was implemented using the *ForceTorqueGroup* node. The *Cochlear* node was changed to expect a *ForceTorqueGroup* in its *stylet* field, which embodied the *Stylet* node itself. These alterations were captured in the *Cochlear* scene-graph (Appendix I, file 'Cochlear_ftgNoLoop.wrl'). Python script (Appendix I, file 'ForceT.py') was generated to monitor force direction and magnitude, moving the stylet along a linear path (by accessing its *translation* field during runtime) when both were at acceptable values. In the *Cochlear* scene-graph, *Frame* and *Button* nodes were used for GUI display of device position and output forces during interaction with the stylet. The *ForceTorqueGroup* node was useful in providing visual and (basic) haptic feedback during collisions, with force quantities output to buttons on the GUI. Stylet removal was realised in the initial

simulator design, however the recoil properties of the array during this process are yet to be implemented. Stylet withdrawal was also implemented via the keyboard, using the key ‘w’ (Appendix I, file ‘keysR3.py’). This may be useful for scenarios where there is only one haptics device used in the simulation and stylet withdrawal functionality requires implementation through the interfacing to another device (such as the keyboard). This is the case in this simulation, where the 3DOF haptics device has not yet been included as the second device for object manipulation and control.

5.2.6 Scene Magnification

Due to the miniature size of the ST and electrode, scene magnification is essential. The surgeon uses binoculars to view the cochlear implantation, at a magnification level of approximately 20 times. However, to maintain haptic accuracy, whilst the visual representation of the objects within the scene is magnified, the physical characteristics must remain at the same scale. That is, only visual magnification may be implemented. The Reachin API *Viewpoint* node is used to provide a framework for this functionality. The *Viewpoint* node has the attributes of a camera [104], which can zoom in and out of the scene, as well as move its position of view. The node is defined within the Cochlear scene-graph (Appendix E, file ‘run_simulation10.wrl’) and the *fieldOfView* field (depicted in Figure 5.5) is initialised to the value 0.03 radians. The smaller the value contained in *fieldOfView*, the larger the appearance of objects within the scene. Python script (Appendix E, file ‘zoom7e.py’) provides the zoom functionality by referencing the *fieldOfView* field during runtime. Its value is increased or decreased depending on whether the user selects magnification (‘+’ key) or demagnification (‘-‘ key) via the keyboard. For this implementation, the buttons must be translated during scene magnification so that they appear to remain in the same position on the GUI.

Using this approach, the user may effectively zoom in and out on the tool, electrode and ST using the keyboard for visual scene magnification while the haptics representations remain at the original (accurate) scale. The maximum magnification level is approximately 32 times (corresponding to a *fieldOfView* 0.0009375) and the minimum level corresponds to visualising the ST near its original size (*fieldOfView* 0.03). There are 6 levels of magnification in total (*fieldOfView*: 0.03, 0.015, 0.0075, 0.00375, 0.001875, 0.0009375). A similar method could be implemented using buttons at the GUI instead of the keys for zoom capabilities.

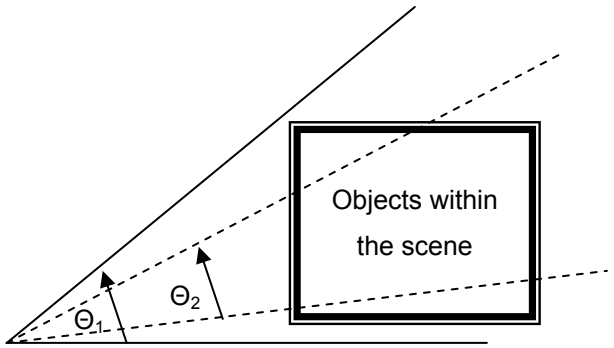

Figure 5.5. An illustration of the *fieldOfView* (θ) field of the *Viewpoint* node. The angle defined by *fieldOfView* is in radians. For θ1 > θ2, objects within the scene appear smaller for θ_1 and larger for θ_2.

Some important design considerations (as well as source of error) are units of scale. The default units of measure for ANSYS and Reachin API are the International System of Units (SI), which are metres for length. The structures designed in this work, including ST, BM, electrode carrier and stylet, are in millimetres and must be converted into metres in the simulation. This is done by setting the *scale* field of the *Transform* node for all structures in the VRML files.

5.2.7 Collision Detection and Response: Stability Issues

Real-time collision detection between the array and ST is a difficult problem. The Contour array has a complex geometry, is made of flexible material that changes in structure (recoils towards the inset electrodes) as the stylet is withdrawn. For the SIT, stylet withdrawal occurs after the array is fully inserted into the ST. The ST also has a complex geometry and may be arbitrarily moved about the scene. As the electrode array is inserted into the ST, it is bounded by the ST walls and the silicone carrier visually, as well as physically, responds to this contact. Collision detection and response between these two objects is therefore non-trivial.

It has been demonstrated that the *CollidingController* node is useful in representing tool tip geometries that vary from the default spherical tip. It enables interaction of the tool tip with objects of complex geometries, such as the ST. However, for complex geometries and tool tips that are comprised of a large number of points, program execution proves slow and is often unstable during object collisions (where forces exceed 8.5N). This results in program termination. Attempts to use this node for direct haptic representation of the electrode array and interaction with the polygonal surface representations of the ST and BM were eventually abandoned. This was after several attempts at program simplification were trialled, including the development of a simplified version of the Contour array in ANSYS (represented by fewer polygons, as shown in Figure 5.6) and haptic interaction with the ST only. The stylet and accompanying withdrawal techniques were omitted, as well as the haptic representation of the BM structure and information display buttons at the GUI.

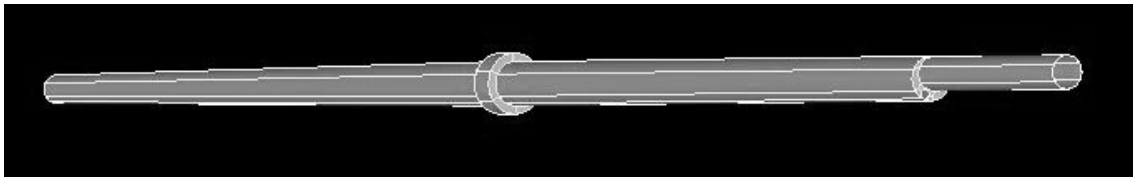

Figure 5.6. A modified version of the Nucleus® 24 Contour™ electrode produced in ANSYS. The structure has fewer polygons representing its surface than its original form.

To overcome the instability issues during haptic loop traversal, due to the large number of collision points comprising the tool tip, and in order to facilitate smooth insertion of the carrier into the ST, the representation of the carrier had to be further simplified. Modification of the electrode model produced in ANSYS was required. Instead of using the polygon representation of the carrier, sample points were strategically distributed along its length. Point positions were determined from the original coordinates in ANSYS. Python script was used to translate, rotate and scale these points in order to obtain an optimal set of point positions in the Reachin API. During this process, the points were superimposed onto a visual representation of the carrier in the Reachin API. For haptic representation of the carrier, the sample points were included in the *Cochlear* scene-graph. The sub-sampled array was of *PointSet* geometry, within the *Shape* node of the *CollidingController*. The same set of coordinates was added to the *points* field of the *CollidingController* node and a haptic representation of the ST in its *CollisionChild*. The entire *CollidingController* node is used in a *Switch* node for selection of the array and stylet using the device button. Device exchange is implemented in 'MoveCarrier_noCA11.py' (Appendix E). In the file 'run_simulation10.wrl' (Appendix E) the sample points are initiated with an empty array. When the user presses the button on the haptics device, the sub-sampled Contour

array is selected and the points are defined along its length (Appendix E, file 'MoveCarrier_noCA11.py'). The new, sub-sampled carrier viewed at the GUI is shown in Figure 5.7. A separate structure comprising the end of the carrier was derived from the original surface representation of the carrier and produced in ANSYS. It simply acts as a visual aid for the surgeon. It does not include the carrier length which houses the electrodes, but includes the marker ribs and end portion of the carrier which the surgeon may 'grasp' during a virtual CI insertion.

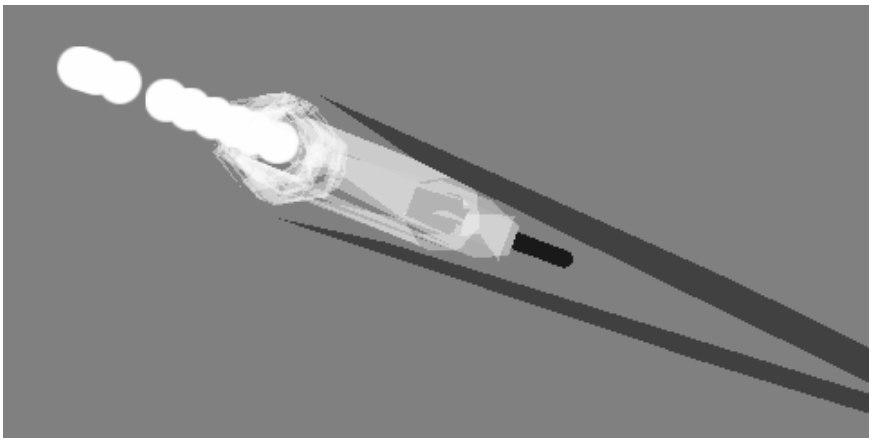

Figure 5.7. The sub-sampled silicone carrier at the GUI, in the Reachin API.

During sample point assignment, it was important to carefully distribute the points so that a separation distance was maintained between adjacent points (overlapping points created excessive force output, greater than 8.5N). However, it was also necessary to define a sufficient amount of points with reasonable dispersion. Whilst this was somewhat subjective in terms of placement, the insertion procedure including array trajectory and electrode design were reviewed so that sample points were located in satisfactory positions, for collision with the ST walls.

Staccato movement of the array and stylet was observed during run-time, as the electrode was selected and arbitrarily moved about the scene by the user. Results indicate that despite the few number of points defining the array (12 points were used in the final simulation), the large number of points comprising the polygons on the surface of the ST (160 *IndexedFaceSet*'s in total), made the collision detection process computationally expensive. Even with one point representing the tip in the *CollidingController points* field, system performance remained less than satisfactory. In effect, visual rendering was slow and the delay in visual rendering of the scene was detectable by the human eye. Collision response between array and ST or BM proved unstable (resulting in erratic avatar motion and/or program termination as output forces in excess of 8.5N were reached). Time delay and proxy penetration of the surface during this time interval may cause such instability. A *ForceTorqueGroup* node enabled indication of force magnitude during the ST and tool collisions. Electrode proxy position and force output during object interactions were shown at the GUI.

5.2.8 Program Optimisations

In order to achieve system stability and real-time interactions without delays in the visual or haptic rendering loops, various optimisation techniques were utilised and the results assessed in terms of accuracy and system response. The first has been described and is simplification of the silicone carrier by sub-sampling the original structure, from its tip to the first marker rib. The second approach considers the reduction in complexity of the ST, with focus on reducing the amount of polygons comprising its surface. Techniques included changing the polygon structure to comprise solely of *IndexedLineSet* instead of *IndexedFaceSet* geometries, producing a mesh over the ST

surface and finally, the application of commercial software and manual methods in a step-wise approach to reduce the number of polygons originally formed in ANSYS.

5.2.8.1 Polygon Reduction of the ST Surface

The large size of the dataset representing the ST surface was significantly reduced by using *IndexedLineSet*'s instead of *IndexedFaceSet'*s to define the physical boundaries of the structure (refer Figure 5.8). File size decreased from 4.07Mb (VRML version 2 file with apical region) to 75.5Kb, with a total of only 15 *IndexedLineSet*'s forming the ST boundaries. This model was ineffective: the haptic representation of the ST was not only inaccurate, but large holes between the splines meant that the proxy was continually falling through the surface. This did not accurately capture carrier/ST collisions and the response was also unrealistic (since the carrier in practice collides with the walls of the ST and not a line set approximation). However, using the line set representation of the ST did produce stable results (program termination was avoided and the movement of the electrode appeared less 'jerky' and hence more realistic). The line set approximation defined the haptic representation of the ST, while the visual representation (which lacked *surface* fields and hence physical representation) was represented by the original ST structure produced in ANSYS. Whilst more lines can be added to the representation to reduce the significance of the holes, the line representation was deemed unsatisfactory for true depiction of the ST periphery.

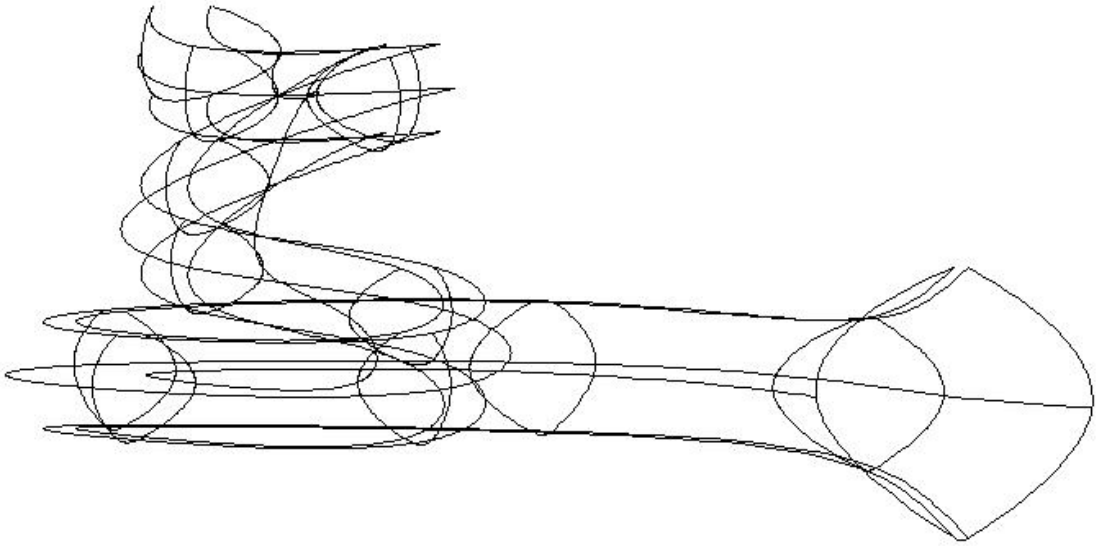

Figure 5.8. A line set (spline) approximation (produced in ANSYS) for the physical representation and haptic rendering of the ST in the Reachin API.

A second, more accurate depiction of the ST walls was derived from the original ST surface (Figure 5.2). Using this technique, the ST surface was replaced with a coarse mesh of polygons in ANSYS. To construct the mesh, an element type (SHELL63) was defined. The *Mesh Tool* (in the *Meshing* sub-menu within the *Preprocessor* menu) was then used, with *Element Attributes* set to *Areas* and a *Smart Size* of 10 (Coarse) was selected. A *Quad* geometry for the mesh was chosen and the *Mesh* procedure activated. The result of this operation was shown in Chapter 3 (Figure 4.14).

Following the routine modifications (VRML 1.0 conversion to VRML 2.0 and modifications for rendering in the Reachin API), further changes included the omission of *Color* nodes and field indexes, as well as the deletion of *normal* fields and related indexes. *IndexedLineSet*'s were also removed, however the *IndexedFaceSet*'s remained. These changes caused a significant reduction in file size: the original file was approximately 5.3Mb and was reduced to 0.5Mb. The meshed surface was substituted for haptic rendering of the ST, which involved the inclusion of surface fields in the

VRML script of the ST structure. A mesh of the BM was similarly produced and visually rendered in the Reachin API. During interaction with the model in the Reachin API, the stylet structure was removed to note the effect of the mesh and file size reduction on rendering performance. The results did not show marked improvement from those produced using the original physical representation, despite the decrease in file size.

A method combining the application of commercial software with manual techniques for file size reduction and polygon minimisation was introduced. The surface structure of the ST, comprised of triangular polygons, is produced in ANSYS as previously described, however the surface areas surrounding the helicotrema are removed using the *delete areas* option in the Preprocessor *Modeling* menu. This is done in order to reduce file size by omitting data that was deemed unnecessary for the purpose of a haptic-rendered cochlear implantation. The ST physical representation therefore has surface areas extending to the 2.5 turn, which covers the maximum insertion depth for the Contour array, of 450.3º (accounting for 410.9º (mean) + 39.4º (standard deviation)) [31]. The surface structure is exported in VRML 1.0 format. The script then undergoes a series of processes, whereby the file size is significantly reduced and the haptic surface descriptions are appreciably simplified. Following conversion to VRML version 2.0 format, the file size is reduced by manual deletion of nodes that are not required in the haptic representation, including *Colour* and *Normal* nodes, and their respective index fields. *IndexedLineSet* geometries are also omitted from the file, with *IndexedFaceSet* geometries remaining. The freeware 'Vwaif' (version 1.1) is used to remove commas and white spaces from the file, to further shrink it. The program is run in MS DOS:

```
>>      vwaif.exe [8] [infile.wrl] [outfile.wrl]
```

where the parameter [8] causes the numbers within the file to be rounded to 8 significant figures.

The freeware program 'VIZup 1.8' (VIZup™) was used for polygon reduction. It decimates the number of polygons comprising the object surface by progressively deleting polygon vertices, though the locations of the remaining vertices are unchanged. The file 'ST_no_apex_vrml2.wrl' (corresponding to outfile.wrl from vwaif) was opened in the VIZup GUI, which is shown in Figure 5.9.

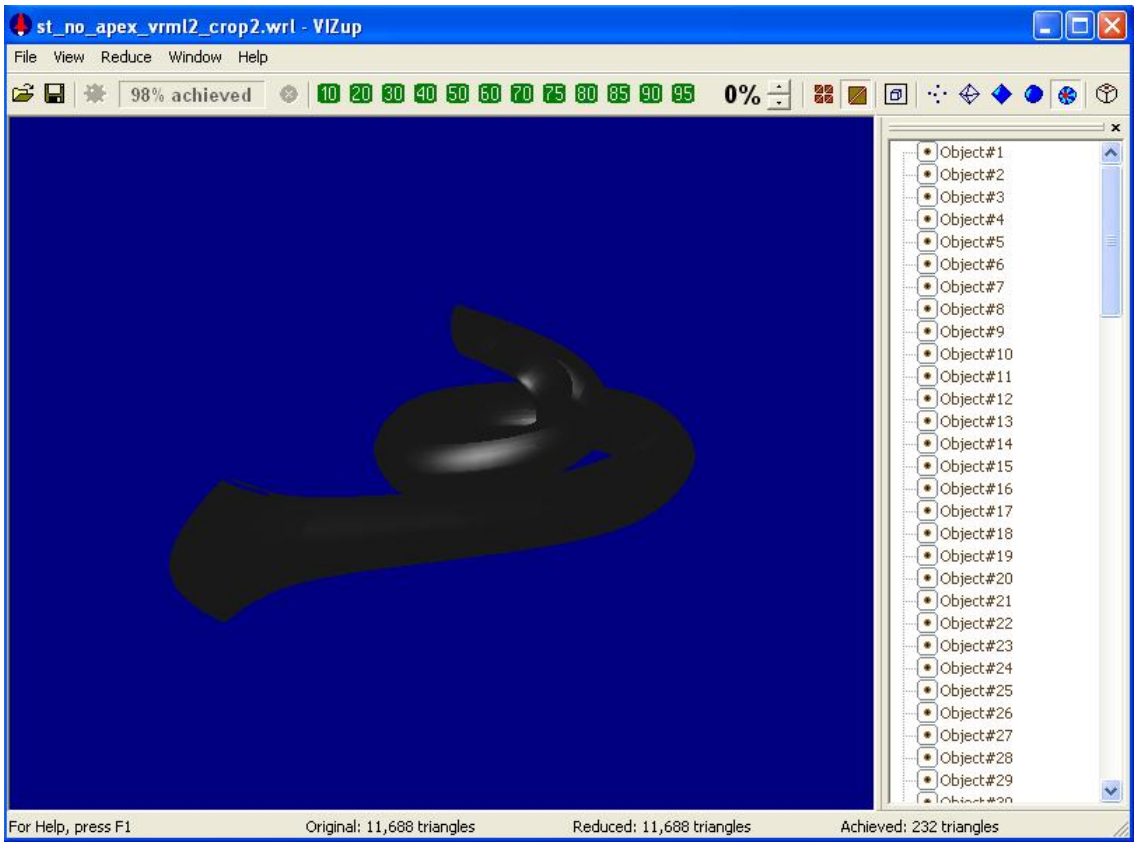

Figure 5.9. The ST shown prior to polygon reduction in the VIZup GUI.

Polygon reduction is initiated upon user selection of the *Start* option under the *Reduce* menu. For the ST representation, the number of polygons is reduced to a minimum of 98%. In this reduction, the ST surface comprised of 232 triangles, whereas

11,688 triangles defined the ST prior to polygon reduction. The results for the triangle reduction percentage, respective number of triangles and file sizes are summarised in Table 5.1.

Table 5.1. Summary of results for the polygon reduction of the ST surface in VIZup.

Polygon Reduction Percentage (%)	Reduced File size (Kb)	Number of Triangles in Surface	Are Holes Visible on Surface?	Stability Review in Reachin API (3 sample points for carrier)
0	994	11,688	No	No: RUNAWAY (1) error Program termination
10	367	10,518	No	No: RUNAWAY (1) error Program termination
20	333	9,350	No	No: RUNAWAY (1) error Program termination
30	297	8,180	No	No: RUNAWAY (1) error Program termination
40	262	7,012	No	No: RUNAWAY (1) error Program termination
50	226	5,844	No	No: RUNAWAY (1) error Program termination
60	188	4,675	No	No: RUNAWAY (1) error Program termination
70	149	3,506	No	No: RUNAWAY (1) error Program termination
75	129	2,921	No	No: RUNAWAY (1) error Program termination
80	109	2,337	Yes	Yes: however, movement is not smooth (staccato motion of tool)
85	88.8	1,753	Yes	Yes: however, movement is not smooth (staccato motion of tool)
90	67.7	1,168	Yes	Yes: carrier collision with wall is stable. Movement is smooth
95	46.6	584	Yes	Yes: carrier collision with wall is stable. Movement is smooth
98	34	232 (minimum)	Yes	Yes: however, carrier falls through holes in surface

Upon analysis of all polygon reduction results produced in VIZup and through interactive inspection in the Reachin API (both via visual and haptic sensing), the result for a 95% polygon reduction (shown in Figure 5.10). To enable rendering in the

Reachin API, the file is slightly modified to include Reachin API specific nodes, such as *Display* and *Transform* nodes.

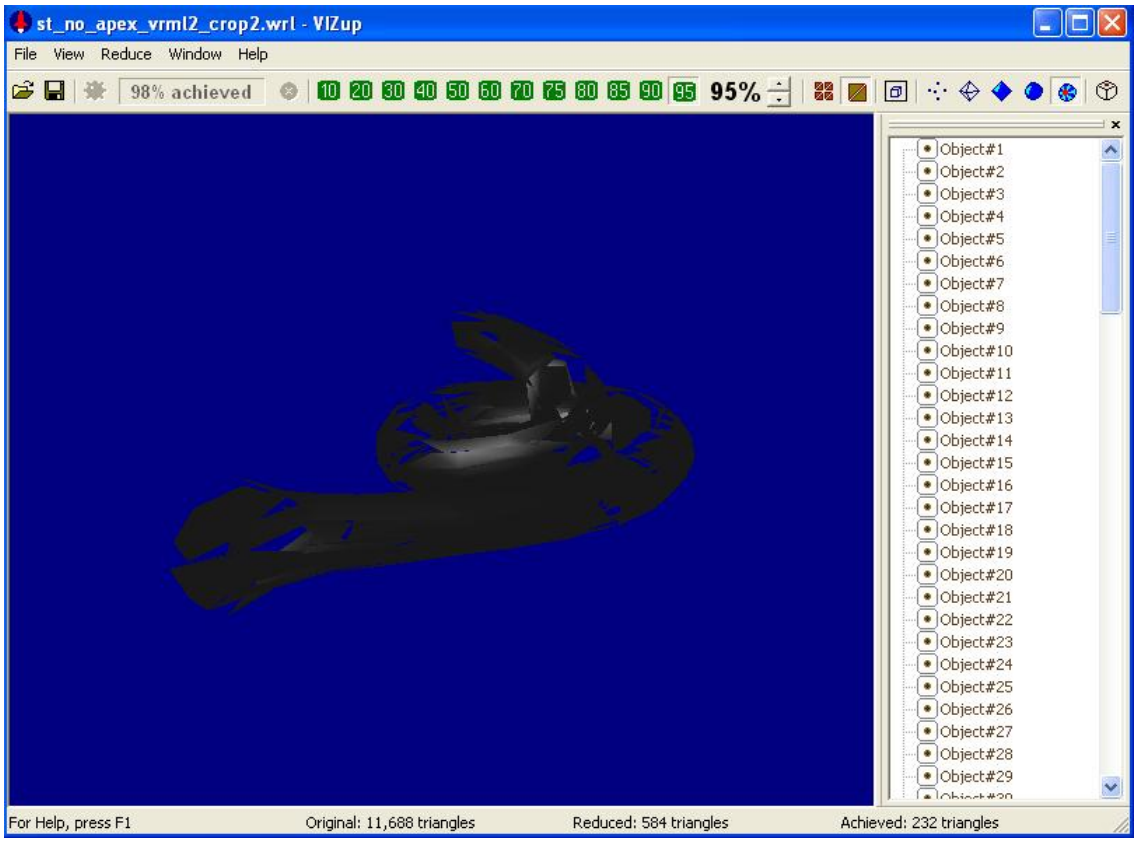

Figure 5.10. The ST surface result for a 95% polygon reduction in VIZup.

This result provides a good compromise between a relatively low dataset size of file size 46.6Kb (compared with the initial dataset size of 3.75Mb for the VRML version 2.0 file 'ST_no_apex_vrml2.wrl' which is without the apical region of the ST) and surface fidelity (including the minimisation of the size and number of 'holes'). Of upmost importance, it yielded a stable result in the Reachin API for the physical representation of the ST, with stable tool and carrier movement, as well as no detectable delay, generation of errors or unexpected program termination for regular movement of the device during collisions. In the final simulation, for visual rendering of the ST the

mesh approximation (Chapter 3, Figure 4.14) was used after file size reduction and modification (application of all stages listed below except step 6.). For haptic rendering, the polygon surface representation of 95% reduction (Figure 5.10) was used, which entailed all stages of the process as described below. This combination provided the best haptic responsiveness, structure integrity and stability, considering the trade-off between real-time system response and ST surface fidelity. Both models are derived from the same original surface structure of the ST, produced in ANSYS.

The final process applied in this work for the derivation of a physical approximation of the ST from the original polygonal surface representation produced in ANSYS is summarised below. The stages of optimisation are listed in sequential order. This process is also applied for representations of the surgical tweezers and Contour carrier end (which was rendered semi-transparent in the Reachin API).

1. The ST apical region is deleted from the haptic representation in ANSYS.

   ANSYS Options: Preprocessor/Modeling/Delete/Area and Below

   This is followed by a manual selection of areas to delete in the ST apical region.

2. The file is exported as VRML version 1.0 from ANSYS.

   ANSYS Options: Plot Controls/Redirect Plots/To VRML File…Replot/OK

   The output is a VRML script in version 1.0 file format.

3. The file is converted to VRML version 2.0 format using the Windows executable file *vrml1tovrml2.exe* in MS DOS.

   >> *vrml1tovrml2.exe infile.wrl outfile.wrl*

   The product is a VRML script in version 2.0 file format.

4. File size is reduced via manual deletion of *Color*, *Normal* and *IndexedLineSet* nodes, as well as any related index fields.

5. Commas and white spaces (including tabs) are removed from the VRML file by running the freeware program *Vwaif*. Outfile numbers are rounded to 8 significant figures.

   >> *Vwaif.exe -8 infile.wrl outfile.wrl*

For haptics representation only

6. Polygon reduction is applied using VIZup.

   VIZup GUI: Reduce/Start

   VIZup GUI: (Select) 95%/Save As/outfile.wrl

   The result with a 95% polygon reduction is used for the physical model of the ST.

****

7. The VRML file is manually modified to include Reachin specific nodes for replacement of some VRML standard 7.0 nodes, such as *Transform* and *Display*. This results in a VRML file (*outfile.wrl*) with a reduced number of *IndexedFaceSet*'s.

8. The *outfile.wrl* is included in *nodename.cpp* within the Reachin API, where the structure is defined. For the ST, this is in the file 'STympV.cpp' (Appendix D).

9. The programs are compiled and the executable is run, as described in section 5.2.2 Dynamic Environment.

   >> *cochlear_node.exe run_simulation10.wrl*

   The output, upon the user zooming in on the scene, is shown in Figure 5.11.

The ST that is visible in Figure 5.11 is the mesh representation used for visual rendering. For aesthetic purposes, the physical representation has been made invisible to the viewer. Hence the viewer does not see two different surface models of the ST. The *transparency* field in the *Material* node was set to a value of 1 within the VRML script (file 'vwaif2_95.wrl', Appendix E) which defines the ST haptic representation. This caused the physical surface properties of the ST to be maintained whilst rendering the object invisible to the viewer. Therefore, the visual model can be seen but not touched whilst the haptic surfaces are tangible but not visible. This reduces the amount of data processed by each rendering loop, since the ST instance is only defined once for the visual loop and once for the haptic loop. The stylet and physical representation of the BM are absent from the simulation in order to reduce the file size and processing time to achieve real-time haptic rendering, although the BM is visually rendered.

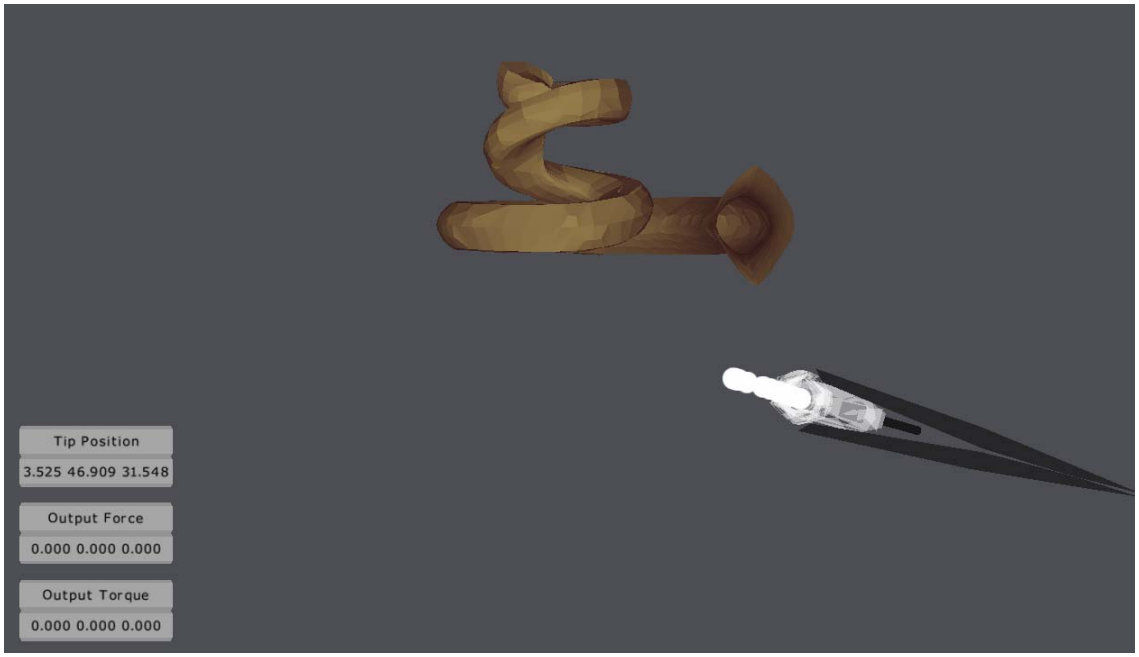

Figure 5.11. The GUI for the simulation in the Reachin API, produced by application of the optimisation techniques for mesh generation (visual) and polygon reduction (haptics). The physical representation of the ST is not visible, only the meshed surfaces for the visual model may be seen.

It was after the process of program optimisation that the number of sample points representing the carrier was consolidated. The maximum number of points that would enable real-time manipulation was established at 12. Beyond this value, system delay was detectable and instability (sensitivity to user-induced manipulations and in some cases program termination) became a detriment to appropriate system functionality. The process of determining the number of points was subjective yet these results were easily judged in terms of system operability and responsiveness. For a simple geometry, such as a *Box*, the number of points representing the carrier could be increased. For this scenario, 33 sample points could interact with the *Box* geometry in real time and without noticeable delay or other instability. When the *Box* geometry was

replaced with the complex geometry of the physical model of the ST, the number of points had to be reduced to a maximum of 12 to achieve similar real-time results. This demonstrates that for more complex geometries, the problem of collision detection and response becomes compounded.

5.2.9 Physical Properties of the ST and Contour Models

To make the simulation realistic, the physical properties of both the ST and the implant need to be established and included in the model. These can be obtained from either measured data or the related literature. For values that cannot be ascertained in this manner, an estimation based on informed judgment should be provided. In this section, the inclusion of the physical parameters into the simulation is discussed. The ST surface is modelled as a *FrictionalSurface* that has the attributes *stiffness*, *damping* and *friction*. The sub-sampled carrier is defined by the *CollidingController* node and the points have physical parameters of *mass*, *proxy size*, *stiffness*, *damping* and *inertia*. The relevant files wherein the physical properties of the structures are defined for the simulation are 'run_simulation10.wrl' for the array and 'vwaif2_95.wrl' for the ST (Appendix E).

5.2.9.1 Physical Properties of the ST

The ST surface is defined using the Reachin API *FrictionalSurface* node, within the *surface* fields of the *Appearance* nodes (file 'vwaif2_95.wrl'). Values for starting and dynamic friction coefficients were obtained directly from measured data and the results for the experimentation are documented in Chapter 3, section 3.5.2. The average starting coefficient for the interface between the silicone carrier and lubricated Teflon surface was 0.0605. This is the value for the field *startingFriction* within the *FrictionalSurface* node. The dynamic friction coefficient for the same interface was determined as 0.0395

and this is the value for the field *dynamicFriction* in the *FrictionalSurface* node. A stopping friction coefficient was estimated as 0.0185 and is included in the field *stoppingFriction*.

Stiffness refers to the capacity of an object to resist a deflection which is caused by an applied force. Damping is a body's ability to reduce the amplitude of its oscillations in order for it to return to rest. For the ST surface, the *stiffness* field refers to the resistance of the surface to a normal force whilst the *stiffnessT* field is the surface repulsion to a tangential applied force. The value for the *stiffness* of the otic capsule could not be found in the published literature and not readily measured for the project, so it is approximated as the stiffness of the human footplate which is composed of compact ivory bone. The mechanical stiffness of the footplate is 441N/m [200] and this the value of field *stiffness* within the *FrictionalSurface* node of the ST. The value for ST *stiffnessT* is estimated at 200N/m. The value for ST surface *damping* is 0.01 for frequencies 1Hz to 100Hz [201]. Human finger tremor (which includes contribution from arm, hand and finger tremor) during object movement is between 1Hz and 30Hz [202], which is within the damping frequency range.

Ideally, values for stiffness and damping in the ST model will be derived directly from measured data. Experiments may be carried out to determine the stiffness and damping using cadaver specimens and/or lubricated Teflon samples, for direct inclusion and comparison in the simulation. Stiffness and damping values for fresh cadaver specimens may yield results that more closely resemble the real operation than the Teflon surface, though this remains to be examined. For this model, only coefficient of friction values were measured using a synthetic environment. Upon further investigation into the actual physical properties of the ST, with time, resources and technology permitting invivo measurements (which currently is not achievable), new

parameters may be included in the ST model. The ease of parameter inclusion into the *FrictionalSurface* node makes the polygonal surface representation for the physical model of the ST an attractive one for simulator progression.

Physical modelling of complex and intricate human structures, where physical access requires maximally invasive surgery, is a challenging exercise and there is little published data on the mechanical properties of the human cochlea. Once technology progresses to a stage where invivo mechanical testing may be achieved with a high degree of accuracy, the parameters may be included in the model. In the current work, values that are closest to the in vivo scenario are used in the absence of quantifiable in vivo measurements or in vitro human data.

5.2.9.2 Physical Properties of the Contour Array

Inertia is a body's ability to resist a change in velocity. An object with a high mass has a high inertia and is more likely to resist acceleration than an object with a low inertia. It is important to define inertia for the carrier, since it is moved arbitrarily (with variable velocity) about the GUI via user manipulation with the haptic device. The inertia of the silicone carrier is defined within the *inertia* field of the *CollidingController* node. A *BoxInertia* node is the closest approximation of the carrier's inertia, since it best represents the behaviour of the implant as it moves about the scene and during collisions with the ST walls. For this type of inertia, the sample points move about the longitudinal axis of the carrier and attempt to return to this position in a spring-damper motion (Figure 5.12). During change in velocity of the carrier, the points act as springs about their 'centre of gravity' (for each point this is defined as their resting position along the longitudinal axis of the carrier). The *BoxInertia* is of a specific size (defining

the boundary and mass distribution of the carrier) and the points have a combined mass which is defined in the *mass* field of the *BoxInertia* node.

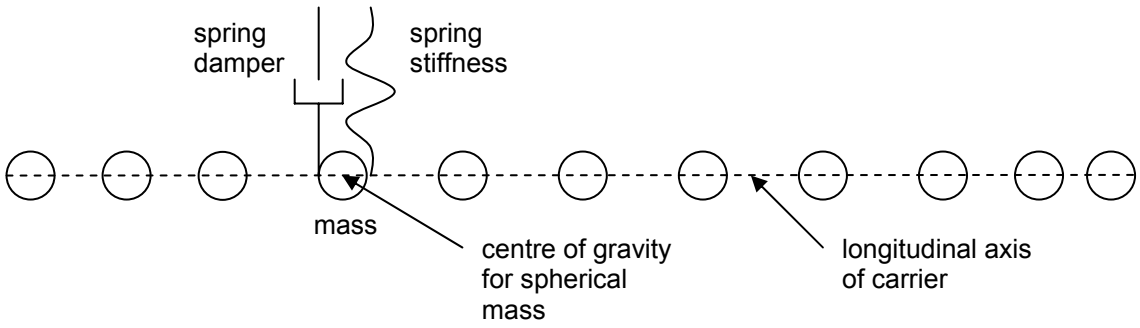

Figure 5.12. Carrier sample points act as springs about the carrier longitudinal axis, on which each point's centre of gravity lies. The 'springs' have stiffness and damping characteristics, for linear (shown) as well as rotational movement.

The mass of a real Contour implant with the stylet in place (from tip to the stylet loop) was measured at 0.0003073 kg. However, the entire tool that the user 'holds' with the device is the combined mass of the carrier and tweezers. Therefore, a pair of surgical tweezers was placed on the scales and the collective mass of the tweezers and implant totalled 0.0178943 kg. This parameter was included in the *mass* field of the *BoxInertia* node and defined as the total tool *mass* field for the *CollidingController* node. As the mass of the object was reduced, it was observed that the carrier's ability to resist change in motion was also reduced.

The size of the *BoxInertia* was subjectively determined by trial and error, and in comparison to observations of real carrier movement (deflections). The value that gave the most realistic movement of the carrier was chosen. The size parameter enables a 3cm movement in the x-plane, a 2cm movement in the y-plane and a 2cm movement in

the z-plane (size 0.03 0.02 0.02). This parameter needed to be defined since its default value is 0 which does not enable deflection about the centre of gravity.

The inertial properties of the carrier change by variation of its stiffness and damping attributes, for both its rotational and linear movement. The fields within the *CollidingController* node that define rotational spring damping and stiffness are *rotSpringDamping* and *rotSpringStiffness*. The default values for these parameters were used: 0.00003N.m.s/rad and 0.2N.m/rad for damping and stiffness respectively. Linear spring damping and linear spring stiffness are defined by fields *springDamping* and *springStiffness*. The field *springDamping* remains at its default value of 1.0N.s/m, whilst *springStiffness* is set to 350N/m. The value of *springStiffness* was determined by considering the Contour stiffness profile documented by Kha et al. [42].

It was observed during program execution that an increase in the values for *rotSpringStiffness* and *springStiffness* resulted in a decrease in the flexibility of the carrier, in relation to its torsional and linear resistance. An increase in the *rotSpringDamping* and *springDamping* caused an increase in the time taken to reduce the amplitude of the oscillating points (when the carrier velocity changes and/or upon collision with the ST walls).

The size of the *proxyRadius*, which represents carrier sample point size, is set at 0.00025m. This is consistent with the actual diameter of the tip (0.5mm in diameter).

The default values for certain physical parameters were used in the simulation, where the real value could not be obtained within the timeframe and/or resources of the project. For future development, values that are obtained experimentally may be added to the model. All physical parameters specified within the work may be easily changed by the developer in the VRML script that represents either the carrier (file ‘run_simulation10.wrl’) or the ST (file ‘vwaif2_95.wrl’).

5.2.10 Haptic Rendering during Full Electrode Insertion

This section of work focuses on haptic rendering during a complete insertion of the implant into the ST. As the electrode array is inserted into the ST, the output force changes as the carrier interacts with the ST walls. In Chapter 3, insertion studies were performed to quantify force delivery during this process. In this section, insertion forces are modelled for the SIT as the virtual carrier is fully inserted into the ST. The technique for synthesising this process is based on redefining the sub-sampled array during collision detection and also relies on carrier proxy position. A program has been written in python to implement this functionality: 'ForceDataCapture29f.py' (Appendix E). Within this script, the proxy position and output forces incurred during object interactions are written to an external file. Data logging occurs during program execution. In Chapter 6, the force and position data obtained from the virtual insertions are compared with experimental results from Chapter 3. Validation of the final system provides an objective measure of simulator accuracy, which is required before it can be introduced into a teaching program for medical education.

An algorithm was designed to replicate the force delivery during insertion of the carrier using the SIT and is implemented in the file 'ForceDataCapture29f.py' (Appendix E). As the carrier is inserted into the ST, the carrier tip position and force output in three dimensions are logged to a data file 'data_logging_all.txt'. The information is appended onto the end of the file, for each set of force and position data. The position data is obtained using the command *dis.devicePosition.get().z* (for the device position in the z-plane) and force information is accessed via *forcetorque.force.get().z* (for force output in the z-plane). The same commands are used to obtain the information in the x- and y- planes. Force output only occurs during carrier and ST collisions.

The direction of implant movement is monitored to see if the electrode is being inserted or withdrawn from the ST. Implant movement along the z-plane represents a change in insertion depth (displacement). If the proxy position (*dis.proxyPosition.get().z*) is greater than the device position (*dis.devicePosition.get().z*), it means the implant is being inserted into the ST (the device is moving along the z-plane 'into' the GUI, which is a decreasing z position). Otherwise, if the proxy position is less than the device position, the implant is being withdrawn from the ST and the device is moving along the positive z-plane (moving 'out' of the GUI). If the implant is moving into or out of the ST and there is a force along the z-plane (*forcetorque.force.get().z!=0*), then the contribution of frictional force is considered and added or subtracted respectively to the z-direction force. The component of frictional force is calculated by obtaining the force acting in the x-plane (which is perpendicular to the z-plane) and multiplying the force along the x-direction by the dynamic coefficient of friction which has been experimentally determined as 0.0395. The force due to friction acts along the z-direction of the insertion (Chapter 3, Figure 3.3) and acts in the opposite direction to the component of input force, F_{in}, along the z-direction yet in the same direction as the output force. The magnitude of the frictional force component is meant to rise as more surface area comes in contact with the ST walls and the force magnitude decreases for the opposite scenario. Therefore during an insertion, the accumulation of frictional force must be added to the z-direction output force and subtracted during an implant withdrawal.

If the force output in the z-direction is non-zero, then the position of the device is considered in order to determine the sample point location. There are 12 sample points distributed along the carrier body from its tip to approximately 15mm along its length. Upon carrier and ST collision around the Basal turn region, the tip of the sample

point is relocated to the second sample point and a point is added to the carrier further along its length, leaving 12 samples defining the physical representation of the carrier. The carrier maintains 12 sample points throughout the insertion. However point locations vary with insertion displacement. At a displacement of 0.6 mm, the tip location changes to the second sample point and another point is defined along its length, leaving 12 samples. This trend continues as the implant is inserted into the ST, as sample locations are defined for dedicated intervals of implant displacement. The sample point number is maintained at its maximum number (12). The additive effect of force due to friction is modelled as the carrier is inserted by adding a force vector as described previously. Therefore the total output force along the z-axis increases as the electrode is fully inserted. The re-sampling of the array enabled the implant to be fully inserted into the ST, resulting in a rise in insertion force along the z-axis (longitudinal axis of the carrier). A visual rendering of the final insertion process is shown in Figure 5.13.

Figure 5.13. The sub-sampled carrier being fully inserted into the ST.

During the insertion process, the output force vectors acting in each of the three directions along with the relative proxy position are written to a log file. This includes linear as well as twisting force information. In total, six forces are logged in the text file against the three dimensions (x, y and z). The proxy position differs from the tip position and represents the carrier tip displacement in the z-direction. During electrode withdrawal, the reverse procedure to implant insertion occurs: where sample point location progresses towards the original tip location. Force and position information are returned to the main function in 'run_simulation10.wrl' for button display at the GUI. The python script is called from this file and the references passed to the *DataLogger* function include the physical model of the ST (*FTSTYMP* node), the *Display* (*DIS*) node, the *CollidingController* (*CON*) node and the *Coordinate* (*PX_COORDS*) node that holds the *point* field of the *PointSet* geometry, with the *CollidingController* node.

The text file containing the position and force results that are logged during a virtual insertion is opened as a Comma Separated Variable (CSV) file in Excel and force in the z-direction is plotted against displacement (z-direction) to show the variation in output force during an insertion. The force profiles show a general increase in output force with displacement for each insertion. The objective of replicating force delivery during insertion of an implant into a virtual model of the human ST has been achieved, yet the simulator must be validated. The accuracy of the results produced from the surgical simulator are examined in the next chapter, in comparison with insertion studies performed using the Contour array in the SIT.

There is one major shortcoming of the method that has been applied for haptic rendering during a virtual cochlear implantation in this work. As the carrier passes the Basal turn region, it does not follow the spiral as it is more deeply inserted. Rather, the sample point positions are redefined in order to permit the points distributed further

from the tip to progress into the ST cavity. The complexity of the electrode movement and the ST geometry make real-time haptic rendering a challenge and this method provided the most stable results as well as replicating force feedback during an apparent insertion. However, further work can be done in this area to provide accurate collision response upon collision detection, correcting the movement of the carrier sub-samples as the silicone deforms during the insertion and without having to redefine sample points to provide a deep implantation. The model and system framework provide a basis on which the developer can implement collision response for the carrier with greater accuracy.

The act of redefining the sample point locations during run-time not only reduces the accuracy of the visual representation of the model and true nature of the trajectory of the array, which hugs the OW during insertion of the Contour array with a SIT, but it does not properly model the true nature of frictional force accumulation. In the simulation, the accumulation of frictional force is modelled so that as the electrode is inserted, there is a higher component of force due to friction. Whilst the number of sample points touching the ST walls increases during initial carrier insertion, after the region of the Basal turn the sample point number defining the array remains the same. The accumulation of force due to friction is replicated by the addition of the force vector. Despite the shortcomings, a real-time haptic rendered surgical simulator for CI surgery has been produced, where the surgeon can see the insertion of the electrode to the region of the Basal turn and beyond this point the user can feel the insertion to maximum depth.

5.2.11 Overall System

The final system for haptic-rendered CI simulation is based on the code in Appendix E, with custom nodes defined in Appendix D. During program execution, the user selects the electrode by pressing the haptic device button and holds the button down whilst inserting the carrier into the ST. As the user inserts the array further into the ST, collisions occur between the carrier and ST walls, causing forces to be reflected back through the haptic device to the user in a closed-loop system. The virtual carrier can be inserted to a maximum depth of approximately 22mm, which is the true insertion depth for the Contour array and is the distance from its tip to the first marker rib [31].

The overall system is summarised in Figure 5.14, from geometric model construction to cochlear implantation. First, surface models of the ST, BM, carrier, stylet and surgical tweezers are created in ANSYS (stage **1.**). The physical model of the BM and the visual and haptic stylet structure are omitted in the present simulation, yet they may be re-included into the model in future developments. A surface structure of the ST is produced and from it, a mesh is formed for its visual representation. A structure for the carrier's end portion is derived from the Contour surface structure in ANSYS. The files are exported and converted to VRML version 2.0 format (stage **2.**). Optimisation techniques are implemented to reduce the file size and decrease the dataset size (stages **3.** and **4.**). Polygon reduction is applied using VIZup to reduce the polygon count that comprises the physical model of the ST. Modifications are made to the file, to enable it to run in the Reachin API by inclusion of specific Reachin API nodes, such as *Display* and *Transform*, as well as including a function call to the python script where object rotation is implemented (stage **5.**). The structures are included in their respective node definitions, within the .cpp files (stage **6.**). The simulation is run from MS DOS (stage **7.**): >> cochlear_node.exe run_simulation10.wrl

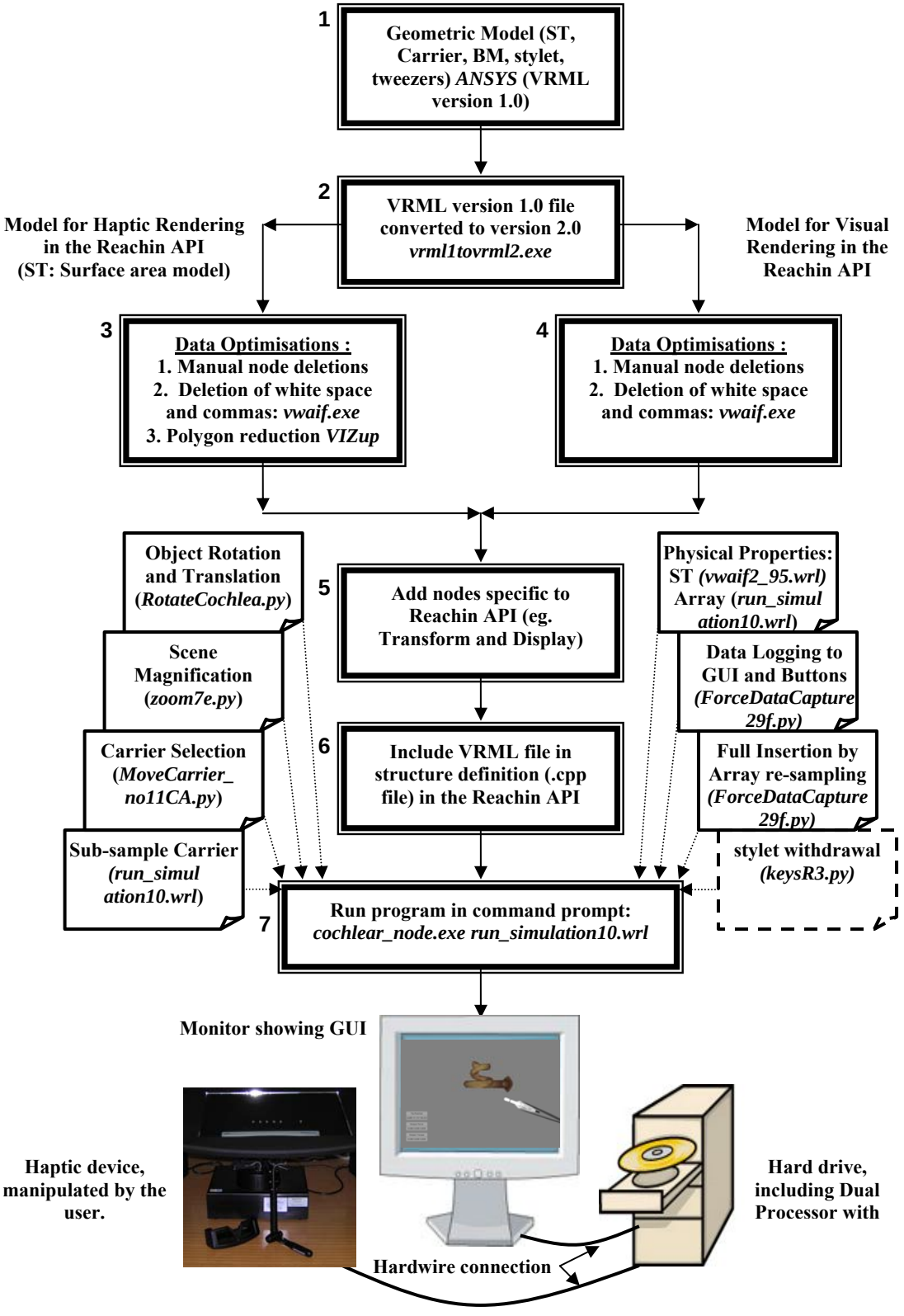

Figure 5.14. Overall flow diagram of the simulator, from geometric model development to virtual cochlear implantation. Program names are in italics.

The script ‘run_simulation10.wrl’ contains function calls to other programs that implement various system functionalities that are controllable by the user. These include scene magnification, object rotation and translation, carrier selection with the haptic device button, as well as the real-time logging of force and position data to a text file and for display on buttons at the GUI. Carrier sub-sampling is defined in a python script and full insertion of the array is realised in a separate python script, by real-time changes in the carrier sub-sample form. The physical properties of the ST and carrier are included to make the simulation more realistic.

Stylet withdrawal has been implemented via a keyboard action, since only one haptic device is used in the simulation. It is not included in the present simulation, however this functionality may be later included, with the capabilities to mimic carrier flexing during stylet withdrawal, as the final stage in the SIT. The BM is visualised in the current simulation (it is the same colour as the ST), however its physical representation is omitted. It may also be included in future designs, by replacing the *IndexedFaceSet* geometries comprising its surface with *MembraneIFS* geometries and including the BM node for its physical structure in the *collisionChild* of the *CollidingController* node. The value in the field *stiffnessM* of the *MembraneIFS* node, that sets the degree of BM deformation, could be determined experimentally or from the published literature.

Upon running the simulation via the command prompt, the user may interact with the virtual environment using the haptics device. Force feedback is provided to the user as the surgeon performs a real-time, virtual cochlear implantation into a surface description of the human ST. Force delivery during implantation of a sub-sampled Contour array is modelled for the SIT. In Chapter 6, the accuracy of the haptic-rendered procedure is assessed using experimental results obtained from the insertion studies.

Chapter 6

SIMULATOR RESULTS AND VALIDATION

6.1 Simulator Force and Position Results

The stages of simulator implementation are followed by its validation. Virtual insertions were repeated to obtain two hundred sets of force and position data, for the analysis of simulator results. The output was compared with results obtained experimentally by fitting splines to the force data and statistically comparing the outcome, at regular intervals along the curves. The statistical analysis involved the determination of the average forces of each complete dataset and plotting the difference of the mean, as well as upper and lower confidence intervals. Ideally, the difference between averages for the results produced from the simulation and via experimentation should be 0N. Any variation about this value is discussed and the spline approximations are analysed in order to review the accuracy of the simulator.

Haptic-rendered cochlear implantation was performed using the surgical simulator to produce 200 sets of force and position data. During an individual insertion, the data was logged in real-time to the external file ‘data_logging.txt’. This occurred for the duration of time that the implant was selected by the user. This was whenever the haptic device button was pressed and held down. For each result, the three-dimensional position of the device tip was given and at that position, force and torque information were documented. Upon opening the ‘data_logging.txt’ file as a CSV file in Excel, one may produce a force profile by selecting the relevant information to plot. In this case, it

was the linear force in the z-direction and device position along the same axis which were of interest. This data corresponds to the force and displacement information obtained experimentally in the insertion studies (as described in Chapter 3). The primary passage of insertion for the simulation (z-plane) and experimentation were therefore equivalent.

It is important to maintain consistency and control variable parameters for the experimental and simulated environments, so that the results of each insertion study can be compared with minimal error. To minimise variability between the simulation and experimentation, the electrode was inserted perpendicular to the x-y plane for all simulation insertions. The tip was advanced approximately 1mm to 3mm into the virtual ST opening before the insertion commenced, as was standard for a typical insertion using the Instron force measurement device. During analysis of force insertion results from the simulation, the process of initial electrode advancement was taken into consideration (since the data was logged prior to the insertion, upon carrier selection).

Despite attempts to standardise the insertions, it is acknowledged that there would have been minor variability between insertions. The disparities are caused by changes in the position of the silicone carrier held at rest in the tweezers, the distance of the initial electrode advancement into the opening of the ST which can vary between 1mm to 3mm (prior to commencing an insertion) and insertion depth. In contrast to the experimental set-up where the electrode is clamped in a fixed position, the user has the freedom to move the carrier about the GUI during a virtual insertion, which will create variability in position and velocity of the carrier between successive insertions. Although it was not possible to exactly replicate each insertion, efforts were made to minimise the variability and regulating the movement of the electrode, similar to a real insertion.

The results were collected and plotted in Excel for all 200 insertions performed in the virtual environment. A typical force profile of the force output in the z-direction versus displacement is shown in Figure 6.1. The sampling rate of the data in the Reachin API is 30Hz for traversal of the graphics loop, within which the collision detection occurs and the force information is written to the output file 'data_logging.txt' from the script 'DataForceCapture29f.py' (Appendix E). This is a higher sampling rate of the data than that applied for sampling in the experimentation (which was 20Hz).

In order to assess the simulation results and for comparison with experimental results, splines were fitted to each force profile produced from the simulation. In its current form, the force data cannot be compared between simulated insertions since each sample point does not occur at a common displacement. The simulated and experimental results could not be directly compared for the same reason. To get average force values at dedicated insertion positions for the simulation data, such as at the Basal turn, a single point needs to be derived from the data at a specific position. Curve fitting was required.

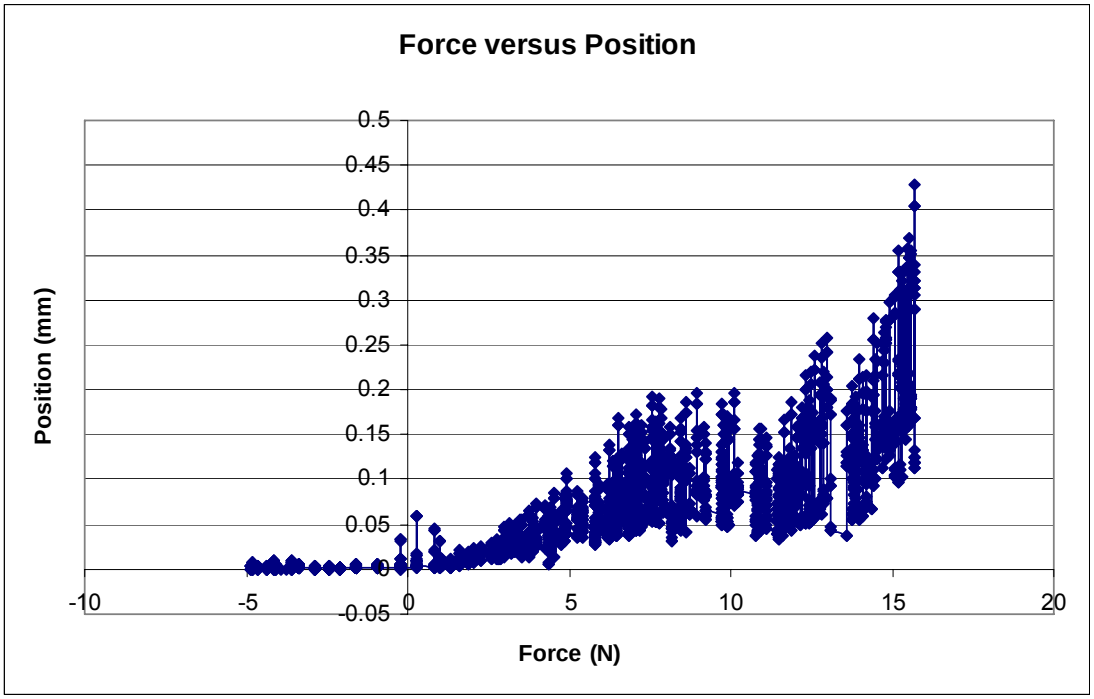

Figure 6.1. A typical force profile produced in Excel, showing output force (N) from a simulated insertion plotted against insertion displacement (mm). The force and displacement are along the z-direction.

6.2 Curve Fitting

Non-linear curve fitting involves the creation of a spline which best represents a set of data points. High order analytic functions are fitted to produce the spline approximation of the data. A range of commercial software programs are available for rapid generation of functions that produce splines of best fit for a given set of data points. Some of the commercial curve fitting and statistical analysis software include SPSS (SPSS Inc.), JMP (SAS), MATLAB (Mathworks, Inc.), GraphPad Prism (GraphPad Software, Inc.) and TableCurve 2D® (Systat Software, Inc.). In this work, each of these software programs was tested to determine the most appropriate for this application. TableCurve 2D was the software of choice, mainly due to its extensive range of functions that are

available for curve fitting and the ease as well as speed with which the user can create complex splines to model the data points. The splines produced in TableCurve 2D could be saved in Excel format. The trial version of TableCurve 2D 5.01 was downloaded from the Systat website [203] and installed. TableCurve 2D has 3,665 equations that can be automatically fitted to a dataset and the functions, including coefficients, are documented for user reference. The GUI of TableCurve 2D is shown in Figure 6.2.

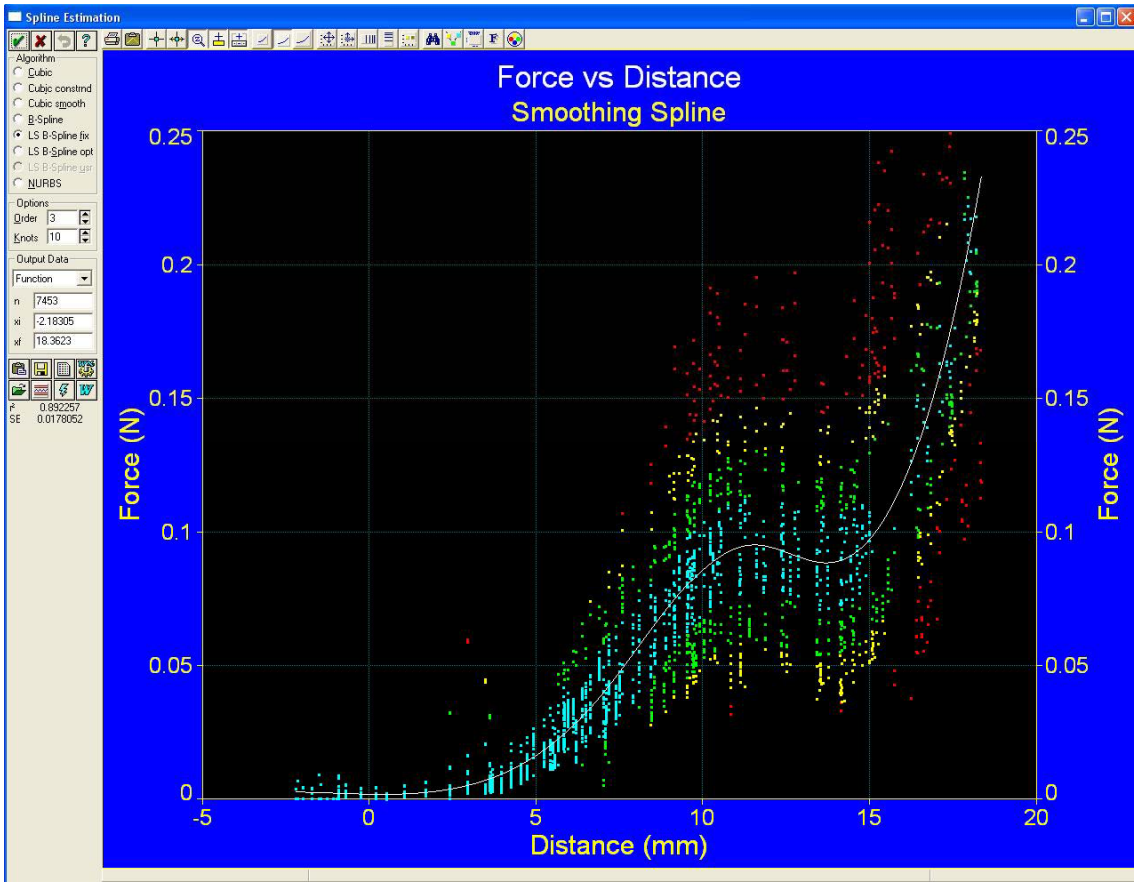

Figure 6.2. TableCurve 2D GUI showing a Least-Squares B Spline fitted to the dataset of points representing output force as a function of insertion distance along the ST.

The *Spline Estimation* menu option is selected in the GUI and the method chosen for spline fitting is the *LS B-Spline fix* which applies least-squares minimisation. The least squares method operates by finding a function that best fits a dataset by minimising the sum of the squares of the residuals (differences of the ordinates) between the data points and the curve. The output of this function is shown in Figure 6.2. Consolidating this option, would convert the spline to the new data series, as pictured in Figure 6.3.

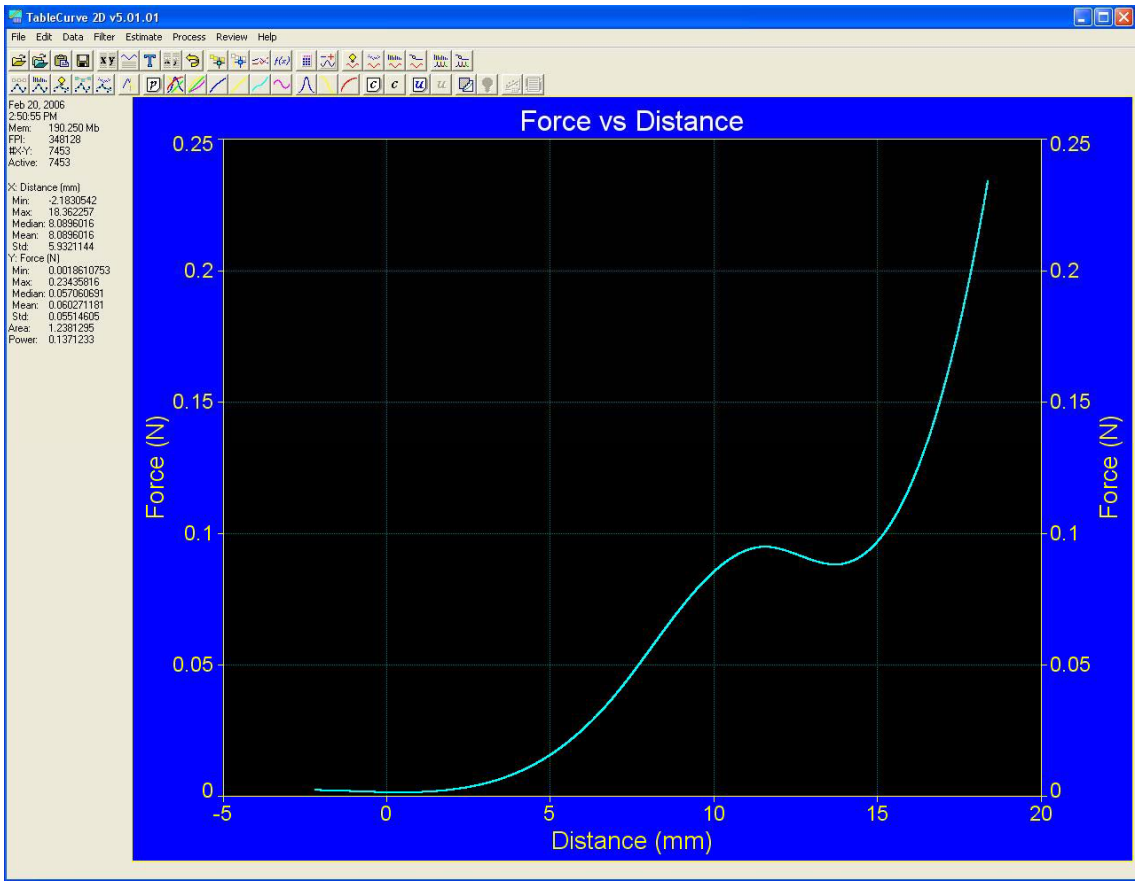

Figure 6.3. TableCurve 2D GUI showing the spline produced by the Least-Squares B Spline option which replaces the original set of data points.

After the spline is fitted to the array of points representing force and position in the z-plane, a number of equations can be automatically generated in TableCurve 2D to numerically describe the behaviour of the spline. To activate this function, the user selects the option *Curve Fit All Equations* under the *Process* menu option. This will generate a list of equations that represent the spline as shown below the graph, in the order of the best approximation (Figure 6.4). Splines and a list of equations defining the curves of best fit were generated for the entire set of data produced from the CI insertion simulations and from the experimentation. For each result, the equations at the top of the list were applied to represent the spline, in order to determine the equation that best represents the entire dataset. It was concluded that a *Fourier Series Polynomial 9 x 2* equation (15) produces the most accurate, universal description of the spline data for all test results.

$$y = a + b\cos(x) + c\sin(x) + d\cos(2x) + e\sin(2x) + f\cos(3x) + g\sin(3x) + h\cos(4x) + i\sin(4x) + j\cos(5x) + k\sin(5x) + l\cos(6x) + m\sin(6x) + n\cos(7x) + o\sin(7x) + p\cos(8x) + q\sin(8x) + r\cos(9x) + s\sin(9x) \quad (15)$$

where ‘a’ to ‘s’ are coefficients that are defined for one particular spline fit and ‘x’ is the sample position at dedicated points along the spline. The values are provided in a separate file within TableCurve 2D for each equation defining the specific spline.

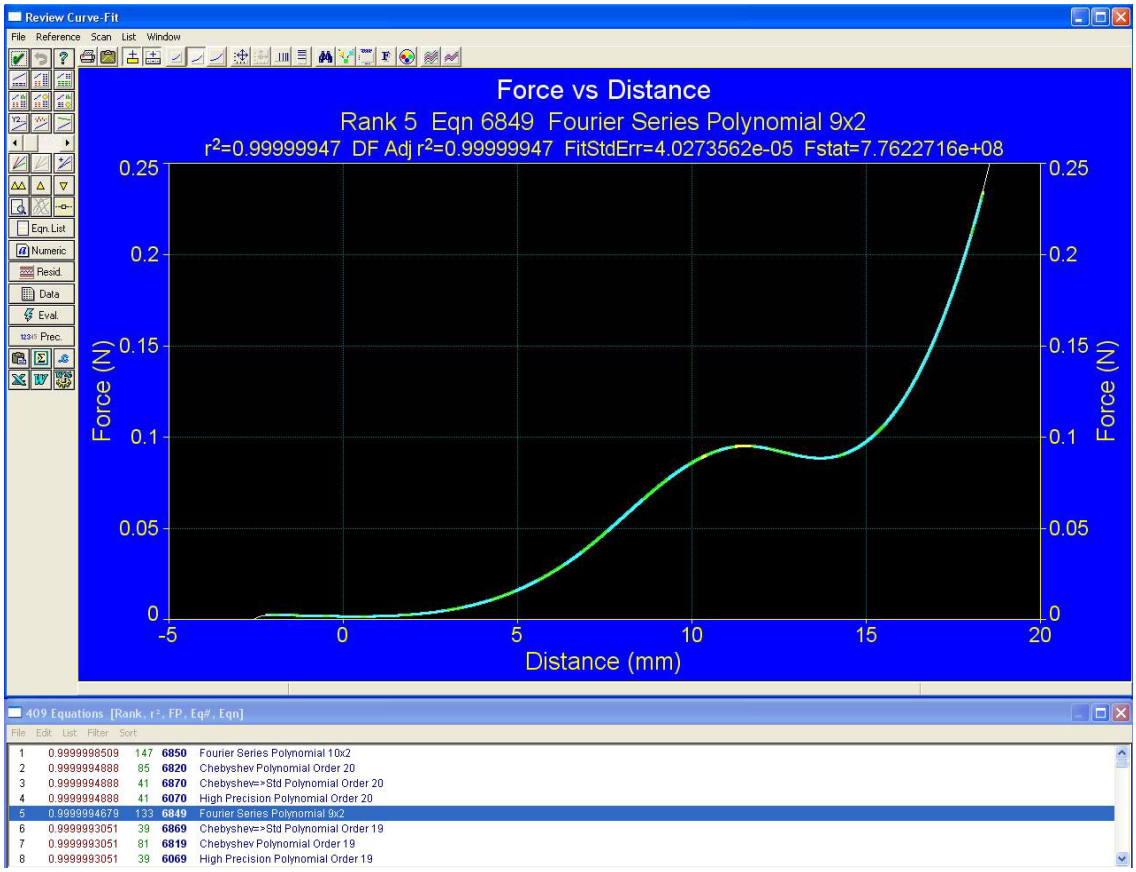

Figure 6.4. TableCurve 2D GUI showing the spline produced by the Least-Squares B Spline option and a list of equations below the graph that represent the spline.

After fitting each spline to the data points using the *LS B-Spline fix* and obtaining the *Fourier Series Polynomial 9 x 2* describing each spline individually, the spline data were saved to an Excel file. The force and position data represented by the spline need to be translated along the z-direction a distance of 2.661mm. This value was calculated by considering the location and magnitude of the ST inner and outer wall radii at the RW, as well as the 3mm maximum electrode advancement prior to starting the insertion. It should also be noted that before commencement of the insertion, the ST is rotated 110º to gain a clear view of the ST opening. A diagram is shown in Figure 6.5 illustrating the relative positions of the ST inner and outer wall radii for determination

of the final factor for spline translation. Once the data is translated 2.661mm, the starting position of the insertion corresponds to a displacement of 0mm, in the z-direction.

Variables†:

Θ = 13.49°, α = 110°

x_1 = 5.54, x_2 = 8.45

z_1 = 1.33, z_2 = 2.02

$v_1 = \sqrt{(x_1^2 + z_1^2)} = 5.697$

$v_2 = \sqrt{(x_2^2 + z_2^2)} = 8.688$

$\beta = \alpha - \Theta + 90°, = 6.500°$

$z_3 = v_1 \cos \beta, = 5.661$

$z_4 = v_2 \cos \beta, = 8.632$

minimum translation $= z_3 - 3$

= 2.661mm

†**all vectors have magnitudes in mm**

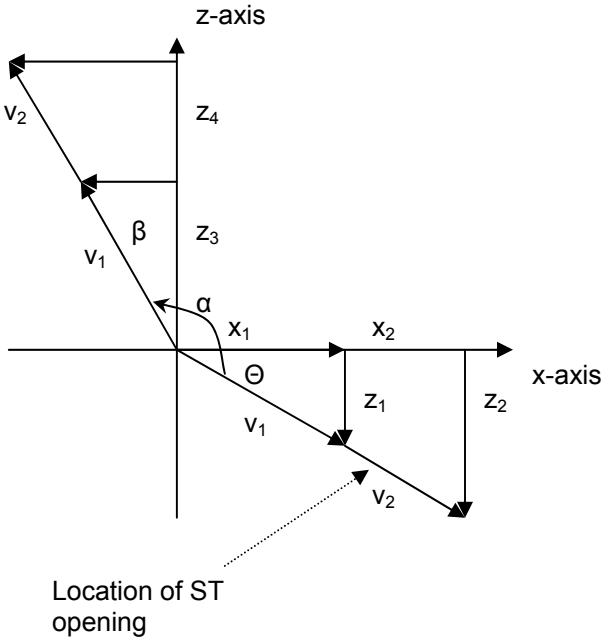

Figure 6.5. The ST opening is translated 2.661mm to make the position of the RW and hence the start of an insertion at 0mm in the z-direction.

After the spline data are translated, the data are sampled at dedicated points along the ST displacement from 0mm to 16.5mm in 0.1mm increments. A function was written in the macro language Visual Basic for Applications (VBA) to sample the Excel data, applying linear interpolation between data points (file 'Acquire Final Stats Sim.xls', Appendix J). The end result is a set of data points that represent insertion force values from 0mm to 16.5mm, with intervals 0.1mm between sample points, as shown in Figure 6.6. This method is applied for the entire set of results (from 200 tests) produced by the virtual cochlear implantations. This allows for statistical data extraction and

comparisons between test results. The sub-sampling method (without point translation) is also applied to the test results produced from the insertion force experiments, for comparison with the force profiles obtained from the simulation and to determine simulator accuracy based on this comparison.

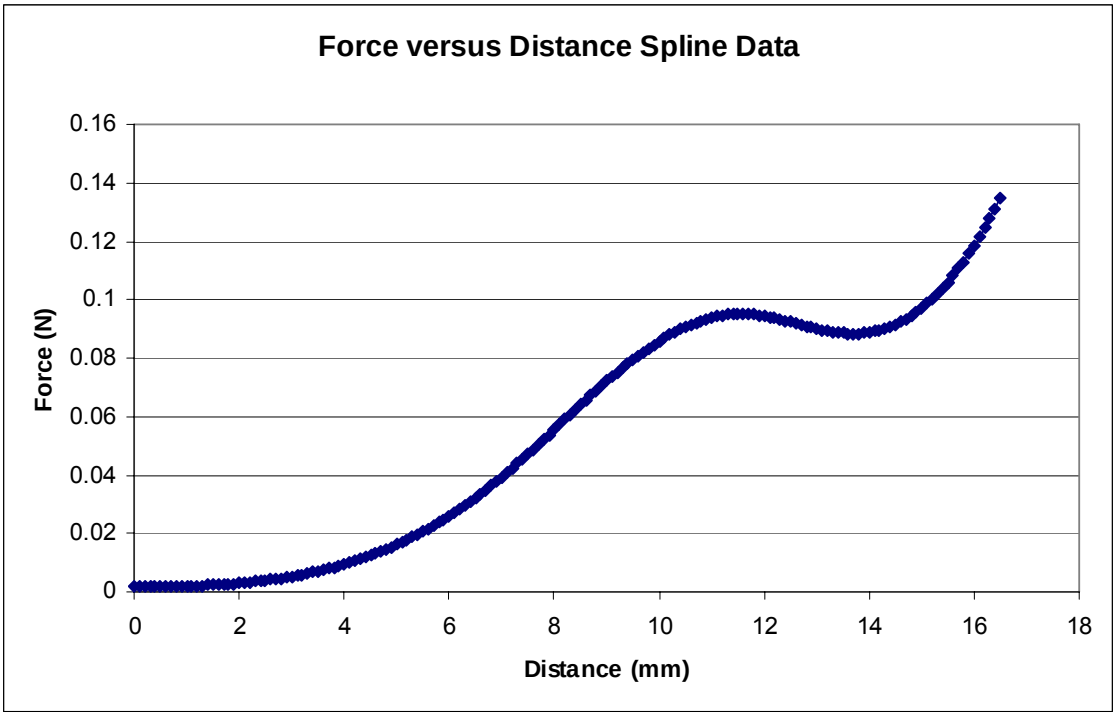

Figure 6.6. A spline representing force versus distance data for a simulated insertion. The sampled curve has been produced in TableCurve 2D and saved to an Excel file where it is translated, sub-sampled at intervals of 0.1mm using linear interpolation between sample points and plotted in Excel.

6.3 Analysis of Simulation Results

Splines representing force versus position data are created, translated and sub-sampled for the explicit purpose of performing a statistical analysis and discussion of results, for the large dataset produced from simulation repetitions. An average force profile was

generated to represent the entire set of simulator results. From this plot, average values with first standard deviations are computed for specific areas of interest, such as the Basal turn. The average force profile is discussed in relation to electrode position and its physical behaviour for the various stages of an insertion.

All spline data approximating insertion forces and relative depth are used to obtain a single spline that represents the average force delivered to the user during a virtual insertion. Average values for output force are calculated in Excel, for an insertion depth from 0mm to 16.5mm at 0.1mm intervals. The first standard deviation is computed using the excel function *STDEV()*. The standard error of the mean is computed by dividing the standard deviation by the square root of the data count (200). Confidence intervals (95%) are determined about the mean by multiplying the standard error of the mean by 1.96. To show the statistical variability of the data, any of the following measures can be plotted about the mean: the standard deviation, the standard error of the mean or the confidence intervals. The average data plotted with the first standard deviations are shown in Figure 6.7.

As the virtual implant is inserted into the model of the ST, the total force generally increases (Figure 6.7). Output force rises during advancement of the electrode into the ST, as the carrier touches the outer ST wall and exerts pressure in this region (Figure 6.7). There is also frictional force contribution as the side of the carrier slides along the outer wall during its advancement towards the Basal turn. Forces increase to an average peak insertion force of 0.095 N ($\pm$ 0.003 N), at a displacement of 11.2mm (Figure 6.7, point 1.). After contact is made at the Basal turn, the output force dips to 0.088N ($\pm$ 0.004 N) at 13.4mm (Figure 6.7., point 2.). As advancement continues past this point, force delivery increases to 0.1389 N ($\pm$ 0.0047 N) at 16.5mm (Figure 6.7, point 3.), which is a total distance of approximately 19.5mm from the cochleostomy site.

Insertion can be continued to the maximum depth of 22mm from the cochleostomy site, however the insertions were stopped around 16.5mm from the starting position (1mm to 3mm inside the ST opening). This is the same stopping criterion as the insertions performed in the experimentation.

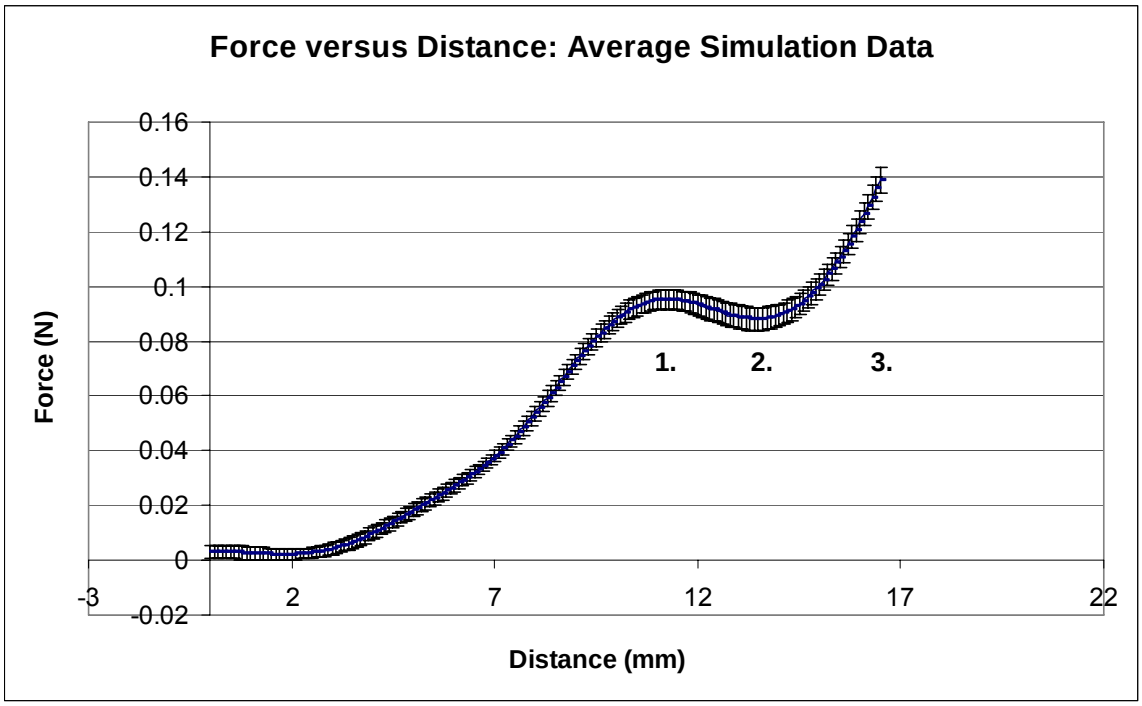

Figure 6.7. The average output force data during a virtual insertion, from 0mm to a depth of 16.5mm, sampled at 0.1mm intervals. The statistics have been computed from 200 sets of insertion data.

Insertion depth is usually 22mm for the Contour electrode [31], which is the measurement from its tip to the first marker rib. However, the displacement of the implant was close to 19.5mm (from the cochleostomy site) for practical insertions. The 2.5mm difference may be due to error in the measurement of the carrier tip from the site of the round window, at its starting position, and may also be due to the carrier not

assuming a perfectly straight form during the experiments but having slight curvature, even with the stylet in place. The final insertion displacement for the simulations was taken as 16.5mm from the carrier starting position so that direct comparisons can be made with the results obtained experimentally, with minimal error.

6.4 Comparison between Simulated and Experimental Results

The results produced from the simulation are compared with the experimental results. First, the two sets of data including the average spline data discussed previously (obtained from the simulations) and the experimental results were reviewed. From this, the general trends of the two datasets were distinguished. Next, a spline was produced that approximated the output forces for each result from the experimental data. The splines were sampled at 0.1mm intervals in Excel using the macro ‘Acquire Final Stats Sim.xls’ (Appendix J). The sampled force data were then combined to give average values of output force over the distance 0mm to 16.5mm (from the initial insertion position to the stopping position). The average output force data with the first standard deviations for the insertions performed using the Instron force measurement device were plotted against displacement in Figure 6.8.

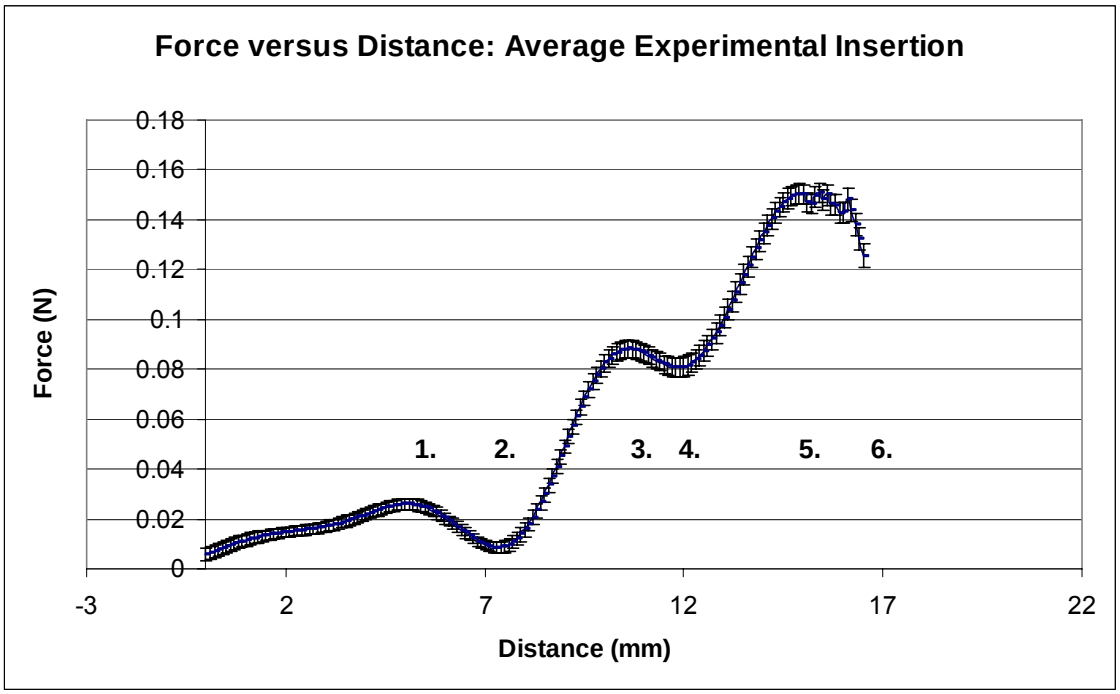

Figure 6.8. The average output force data during a physical (experimental) insertion, from 0mm to a depth of 16.5mm, sampled at 0.1mm intervals. The statistics have been computed from 11 sets of insertion data.

The same statistics were calculated for the experimental data, as for the results produced from the simulation: the standard deviation, the standard error of the mean and 95% confidence intervals. The force profile shown in Figure 6.8. is an approximation of the experimental data which closely matches the results discussed in Chapter 3. The overall increase in output force with displacement, as well as locations of peaks and troughs, remain the same. The peak in insertion force around the Basal turn, which corresponds to the electrode tip touching the lateral wall in this region, measures 0.0880N (± 0.0186N) for the Contour array, at distance 10.6mm (Figure 6.8, point 3.). The average peak value at the Basal turn was measured previously at 0.095N (± 0.0132N), between 9mm and 12mm. The average peak insertion force for the original

data is 0.113N (± 0.0133N), just prior to reaching the peak value of 0.194N (± 0.0287N) where the first marker rib touches. After this point, the force drops off slightly. For the spline approximation, the insertion force increases to a final average peak value of 0.1509N (± 0.0189N) (Figure 6.8, point 5.) before a slight decrease in output force (Figure 6.8, point 6.).

The results produced from the spline approximations of the simulation and experimental data are compared, and are represented by Figures 6.7 and 6.8 respectively. For both approximations, the output force generally increases as the distance traversed by the electrode increases, from its starting to stopping position. The depth of 0mm corresponds to the starting location and a displacement of 16.5mm is the stopping position for the implant. This distance is measured along the z-direction for the results produced from the simulation.

For both sets of results (simulated and experimental), the average output force generally increases to a peak value around the Basal turn region: 0.095N (± 0.003N) at a displacement of 11.2mm for the simulation results (Figure 6.7, point 1.) and 0.0880N (± 0.0186N) at distance 10.6mm for the experimental results (Figure 6.8, point 3.). For the results produced from the experimentation, there is a small peak around the 6mm mark (Figure 6.8, point 1.) and a dip around the 8mm mark (Figure 6.8, point 2.). The force then increases to a peak at the Basal turn (Figure 6.8, point 3.). The results produced by the simulations do not show the same dramatic changes about the 6mm and 8mm displacements as demonstrated by the experimental results. After the Basal turn region, both force profiles (Figures 6.7 and 6.8) exhibit a dip in output force: 0.088N (± 0.004N) at 13.4mm for the simulation results (Figure 6.7, point 2.) and 0.0807N (± 0.0219N) at 11.8mm for the experimental results (Figure 6.8, point 4.). After this point, forces increase in both scenarios to a value of to 0.1389N (± 0.0047N) at 16.5mm for

the simulation results (Figure 6.7, point 3.) and 0.1509N (± 0.0189N) at 15.4mm for the experimental results (Figure 6.8, point 5.). After reaching a maximum value of 0.1509N, the force output for the experimental results decreases to 0.1254N (± 0.0385N) at 16.5mm (Figure 6.8, point 6.).

The degree of similarity (or difference) between the results produced from the simulations and experimentation must be quantified. This will help to determine the simulator accuracy and any anomalies may be explained with reference to the insertion process. The average force values from Figure 6.7 are subtracted from the average force values in Figure 6.7 to give the difference between mean simulator and experimental results, for each 0.1mm interval from 0mm displacement to 16.5mm. The difference in the standard error of the mean, se(diff), is calculated by application of (16):

$$se(diff) = \sqrt{(se(mean\_sim_i)^2 + se(mean\_exp_i)^2)} \quad (16)$$

where se is the standard error, $mean\_sim_i$ is the average value of the simulator results (Figure 6.7) at position i and $mean\_exp_i$ is the average value of the experimental results (Figure 6.8) at position i. The value i varies from 0mm to 16.5mm in 0.1mm increments.

The difference in the standard error of the mean, se(diff), is multiplied by 1.96. This value is then added and subtracted from the difference in the average value ($mean\_exp_i - mean\_sim_i$) to give the upper and lower confidence intervals, as described by (17):

$$CI_{95} = diff +/- 1.96 * se(diff) \quad (17)$$

where CI_{95} is the 95% confidence interval for upper and lower limits, se is the standard error and diff is the difference in average force values ($mean\_exp_i - mean\_sim_i$). The upper and lower levels defining the CI_{95} may be found by using the operators for addition and subtraction respectively, in (17). These intervals denote that there is a 95% confidence that the results will fall within these upper and lower margins. That is, 95%

of the time, the interval between the upper and lower confidence levels will cover the difference in measured force, obtained either by simulation or via experimentation. Figure 6.9 shows the differences in the average output forces between the simulator and experimental results, for the spline approximations. The upper and lower confidence levels are included.

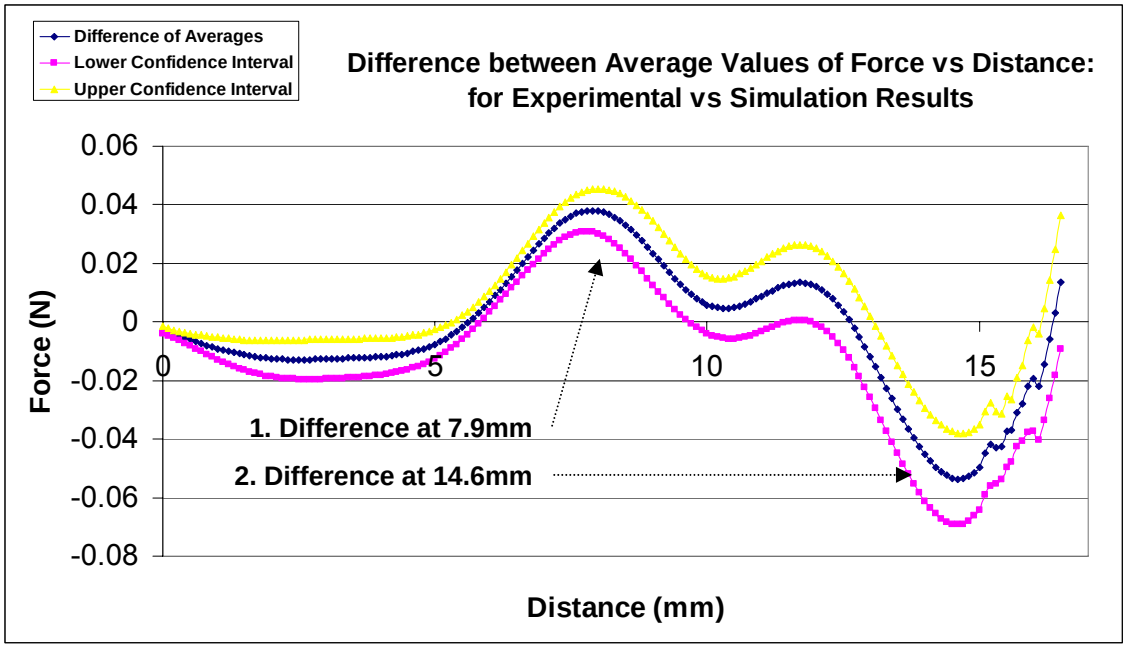

Figure 6.9. The differences between average values for force output from the experimental and virtual insertion studies. These differences are shown in blue ('Differences of Averages'). The upper and lower confidence intervals for the differences in average force delivery are shown in yellow and magenta respectively.

Figure 6.9 shows the differences in average output force between the two sets of results. Negative force values indicate that the simulation force output is greater than the experimental force output, whilst positive values imply the opposite logic. A difference of 0N is an optimal result, as it signifies that the average output force from the

simulation equals the average force from the experimental results at that particular insertion displacement. As the difference moves further away from 0N, the similarity between results diminishes as there is a disparity in force output for the two sets of data. The confidence limits vary about the difference, but at any given displacement these levels indicate that with 95% confidence the difference will be no less than the lower confidence limit and no greater than the upper limit. Where a confidence interval includes 0N, the output force data are consistent with the difference of 0N and a difference has not been established. For this case, the results are not statistically significant and the difference is likely to be 0N. If the confidence interval does not include 0N, the data are not consistent with equal average force values and a difference has been established. In this case, the results are statistically significant and the difference is unlikely to be 0N.

To validate the surgical simulator, the differences in the averages of output force between the two datasets and their confidence levels are analysed to first quantify the variation and then determine the likely causes for the variation about 0N. The implications of the differences in average forces and the respective confidence intervals are evaluated in terms of statistical significance or practical importance. Practical importance in this context means that the difference in force output is high enough that the simulator is considered to be a misrepresentation of the force reflection experienced in the practical scenario (test rig) during an implantation and the information relayed to the user is misleading. Inaccurate force reflection provided by the simulator can lead to practical implications for the training otologist, which could affect the successful insertion of a live implantation, such as excessive force delivery causing damage to the implant or ST structures including the BM, OSL and Reissner's Membrane. It means that the force delivery is misrepresentative of the true scenario and may have an impact

on surgical technique. The regions of major difference where there may be practical implications are identified. Suggestions are made as to the possible methods that could be applied to bring the values within these regions of considerable difference closer to 0N, to improve similarity between practical and simulator results, and achieve greater simulator accuracy.

As the electrode is inserted into the ST from 0mm to 5mm, the difference reaches -0.01285N at 2.5mm (Figure 6.9). At this point, there is a 95% confidence that the difference is no less than -0.0196N and no greater than -0.0061N. Differences in this region can be attributable to the carrier shape: in the simulation the carrier is sub-sampled by spheres of uniform geometry, centred on the longitudinal axis of the carrier which is linear. However, in the experimentation, the carrier form may be slightly curved (not ideally straight) and is tapered. As the carrier is inserted, its outer side touches the cochleostomy site and creates a change in force profile at this point. In effect, force output changes due to a difference in the nature of the ST and carrier interactions between the two scenarios.

The difference between the force magnitudes decreases to 0N at an insertion displacement of about 5.6mm and there is a 95% confidence that the difference will be no less than -0.0035N and no greater than 0.0043N. This means that at this stage where the carrier is sliding along the outer ST wall at a distance of 5.6mm, the force output from the simulator is equal to the force output in the experimentation. Just before the electrode reaches the Basal turn region, the difference increases to a peak value of 0.0380N at 7.9mm (shown by arrow 1. on Figure 6.9), where there is a 95% confidence that the difference in output forces will be no less than 0.0306N and no greater than 0.0454N. In the experimental results, there is a dip in output force at this point due to the carrier tip no longer making contact with the ST inner wall. The tip should, under

ideal conditions, rest against the outer wall since the carrier design with the stylet in place is straight. However, in practice a slight curvature in carrier design was observed during insertion tests which was not replicated in the simulation. The silicone curvature could be caused by human or mechanical influences, most likely the stylet is not fully inserted into its silicone envelope, or deformation of the stylet and/or carrier where either or both are slightly bent. The tendency of the silicone carrier to assume its pre-curled state may add to this effect.

The difference decreases to 0.0046N at 10.3mm, with a 95% confidence that the difference will have a minimum value of -0.0056N and a maximum of 0.0148N. This position corresponds to the location of the Basal turn, in the proximity where the tip would touch the lateral outer wall of the ST. Here, the difference is close to 0N and the confidence intervals include 0N, indicating that at the Basal turn, the simulation closely represents the physical interactions that occur between the electrode and ST walls in the insertion experiments. After the Basal turn region, there is a slight increase in the difference to 0.0135N at 11.7mm, with 95% confidence limits of 0.0005N and 0.0264N. The difference returns to 0N at 12.6mm. As the electrode displacement increases to 14.6mm, the difference in average reaches -0.0536N (shown by arrow 2. on Figure 6.9), where the force output from the simulation exceeds the experimental average with a 95% confidence interval of -0.0693N to -0.0380N. The difference returns to 0N just after a 16.3mm displacement, where the electrode is close to its full insertion depth. At full insertion depth (16.5mm) the difference is 0.0135N, yet the confidence limits, -0.0093N and 0.0363N, reveal that at full carrier displacement the difference is not statistically significant.

The results shown in Figure 6.9 are statistically significant for electrode displacements from 0mm to 5.3mm, 5.9mm to 9.6mm, 11.5mm to 11.8mm and 13.1mm

to 16.1mm. In all other regions, including the location of the Basal turn where the electrode tip contacts the lateral outer wall of the ST and at full electrode displacement, the differences are not statistically significant. For regions that are statistically significant, the difference in average output force does not exceed 0.0536N in magnitude. This is the maximum amplitude of the differences at an electrode displacement shown by arrow 2 in Figure 6.9. Differences are below 0.02N for all statistically significant regions, except for the displacements from 6.6mm to 9.1mm and 13.3mm to about 16.1mm, which include the extremities of the differences indicated by arrows 1. and 2. in Figure 6.9.

Results may be statistically significant but may or may not be practically important. Ideally, the difference between simulator and experimental results should be 0N for all insertion depths. However, for regions in which the difference in average force output is not equal to 0N and that are statistically significant (the confidence intervals do not include 0N), it must be determined whether the difference will have practical implications. That is, whether the difference in average force output is of practical importance. A human can distinguish the difference between two forces if there is a variation in force magnitude of 0.5N [204]. The greatest difference in the magnitude of force between the simulator and experimental results was 0.0536N. The user should not be able to detect a difference in force at this scale. In terms of human haptic perception, the difference is of no practical importance yet is considered statistically significant.

Differences of up to 0.0536N may have practical implications concerning other aspects of the insertion, such as the physical behaviour of the implant and/or cochlear structures. Variation of this magnitude could affect the position and physical behaviour of the implant, such as the magnitude of its deflection during contact with the ST walls,

or may induce varying degrees of damage to the OSL or BM during a live implantation. The magnitude of forces that will directly cause damage to these delicate structures within the cochlea has not, however, been documented in the related literature and future studies may quantify this relationship.

The results produced from the simulations and from insertion experiments show similarities as well as differences. Figures 6.7 and 6.8 show a general increase in average output force as the electrode is advanced into the ST. There are similar peaks in insertion force around the Basal turn area where the electrode tip exerts pressure on the lateral outer wall of the ST. Output force shows a slight decrease after the Basal turn area and then increases to a final peak value. The difference between the averages of the two datasets is shown in Figure 6.9 and varies about 0N. Confidence intervals also vary about 0N and there are regions of statistical significance as well as regions that are not statistically significant. For regions of statistical significance, it is of concern whether the difference in average output forces will have practical implications. For this application, the variability in output force does not exceed 0.0536N and for the majority of regions that are statistically significant, the difference is below 0.02N, which is well below the level of 0.5N at which humans can tactually perceive differences between two forces in haptic feedback. However, there may be implications that are of practical importance concerning the movement of the implant (its position and physical reaction to a collision) and behaviour of other structures surrounding the ST walls that may be later included in the simulation, such as the BM and OSL that are prone to damage during an implantation.

Figure 6.9 shows that around the Basal turn region and near full insertion depth, the differences are minimal and the insertions are similar in terms of output force magnitude. The regions of greatest difference and primary concern are at insertion

depths of 7.9mm and 14.6mm, where the magnitude of difference reaches 0.0380N and 0.0536N respectively. A major cause of discrepancy between the results is attributed to differences in electrode design between the sub-sampled approximation of the virtual carrier and the real implant which pertains slight deformation. As the electrode is inserted into the ST, its design will affect its trajectory. The precision of modelling the trajectory of the implant as it spirals around the ST wall (primarily the outer wall for the SIT) may be enhanced by implementing further changes in implant design during an insertion. Currently, the position of the sample points progress along the carrier away from the tip. Ideally, the sample points should assume a location along the ST wall as the implant advances past the Basal turn and to its final insertion position. The collision response for this task is complex and would involve the determination of the point positions about the tip after a collision has been detected, as well as implementing the functionality by changing the *CollidingController* form in real-time. The latter has been performed in the current work and the template would allow for the modifications described. In order to improve the accuracy of the simulator and minimise differences in force output between the simulation and experimental results, future work should therefore focus on predicting and capturing changes in implant form. This may be done by improving the real-time collision response algorithm and the criteria for determining the *CollidingController* point locations, which should be based on the collision response for the sample points as currently implemented in the file 'ForceDataCapture29f.py' (Appendix E).

Simulator accuracy has been examined by approximating the average output force for the simulator and comparing this with a spline approximation of the experimental results. The splines are statistically compared to quantify the differences in output forces between the two sets of data and the differences with confidence levels are

summarised in Figure 6.9. It has been determined that the differences between the output forces for each insertion method are minimal and for regions of statistical significance, the differences are not of practical importance. Simulator accuracy may be improved by reviewing the collision response algorithm that determines the carrier sample point location and implementing real-time changes in carrier form for its complex insertion trajectory along the ST walls. Further, more insertion trials should be performed in the experimental rig to confirm the results. However, the output force profile generated during a real-time simulated insertion closely resembles the insertion force profiles produced from comparable studies performed via experimentation.

The insertion force profiles produced from the experimentation and simulation are combined into one force profile that approximates the overall electrode insertion in terms of output force as a function of implant displacement along the ST (Figure 6.10). The approximation shows a general increase in output force with insertion depth. At the Basal turn, there is a peak in the output force of 0.0910N at 10.8mm followed by a slight dip to 0.0871N at 12.0mm. Forces then rise to a maximum value of 0.1373N at an electrode displacement of 16.1mm. At a 16.5mm electrode displacement, output force is 0.1322N.

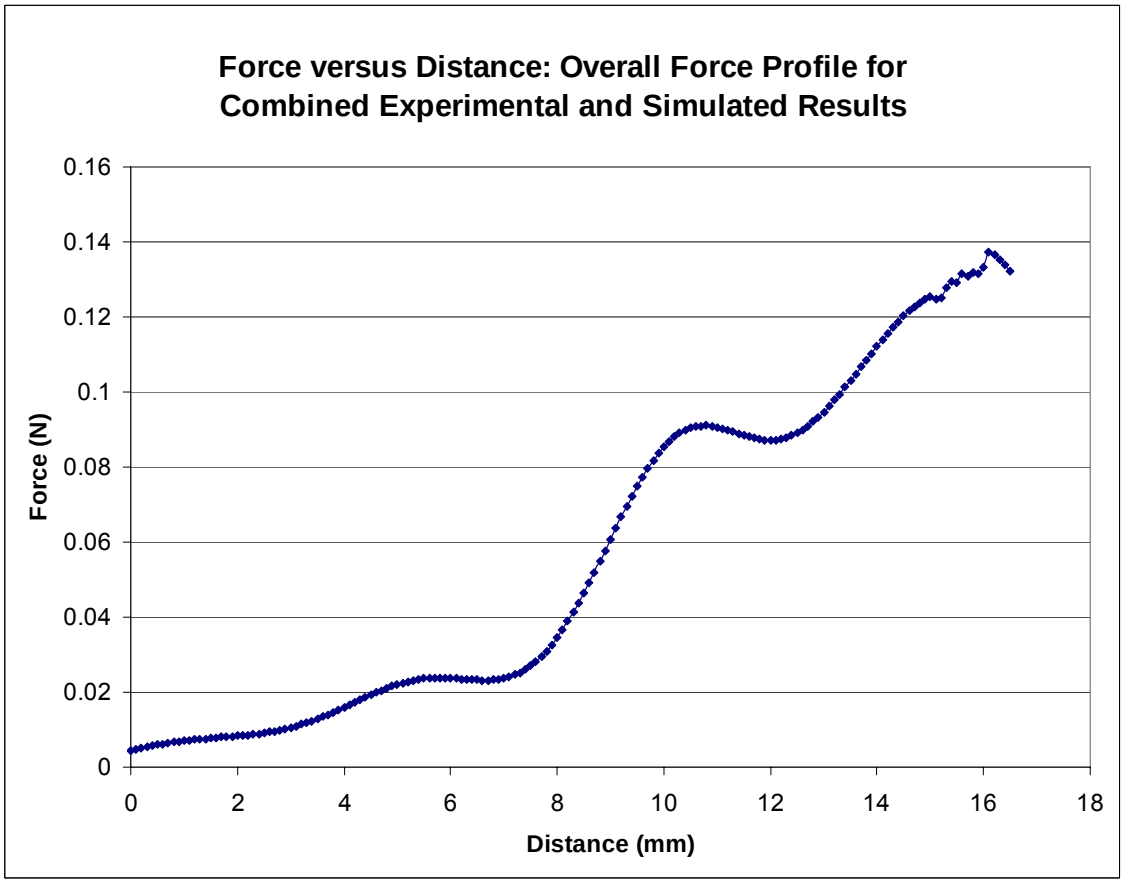

Figure 6.10. The average output force profile for an implantation which is produced by combining the datasets from the experimental and simulator results.

The combined data serves the purpose of producing a single equation to represent a typical force profile of an insertion, produced either by experimentation or simulation. The data was combined in Excel by simply calculating the average output force over the displacement 16.5mm for the two datasets. The data shown in Figure 6.10 is then opened in TableCurve 2D and a curve is fitted to the data by selecting the *LS B-Spline fix* option, which produces a spline that approximates the data produced in Excel. The spline approximation is shown in Figure 6.11. It should be noted that some degree of error is introduced by approximating the data using a spline, however, it is assumed to be a line of best fit and the error to be minimal.

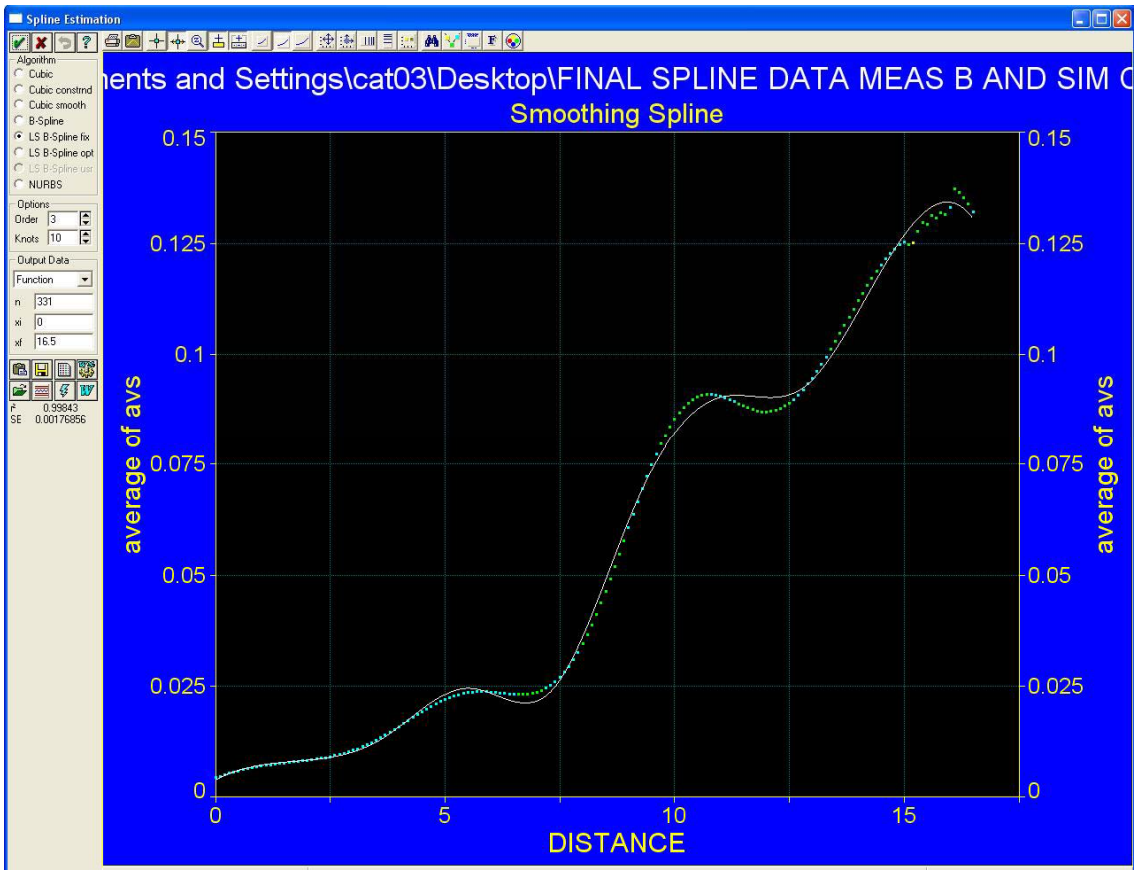

Figure 6.11. TableCurve 2D GUI showing the final spline approximation of the combined dataset of the experimental and simulator results superimposed onto the sampled data from Figure 6.10.

The spline approximation also shows a general increase in output force with insertion depth, as in Figure 6.10. However, the peak at the Basal turn is 0.0908N at 11.35mm (which was previously 0.0910N at 10.8mm). After this point, there is a slight dip to 0.0904N at 12.1mm (previously 0.0871N at 12.0mm). Forces rise to a peak value of 0.1344N at 15.95mm (previously 0.1373N at 16.1mm) and at 16.5mm, the output force is 0.1308N (previously 0.1322N). Following the spline approximation, a list of equations is generated and the *Fourier Series Polynomial 9 x 2* equation selected to represent the spline (as described by (15)). The curve representing (15) is shown in

Figure 6.12, as fitted to the data in Figure 6.10. The parameters that define (15) for this curve fit, including coefficients a to s, are provided in Appendix K (from TableCurve 2D). The function 'y', including its confidence intervals, is defined for values of 'x' from 0mm to 16.5mm in 0.05mm increments (Appendix L).

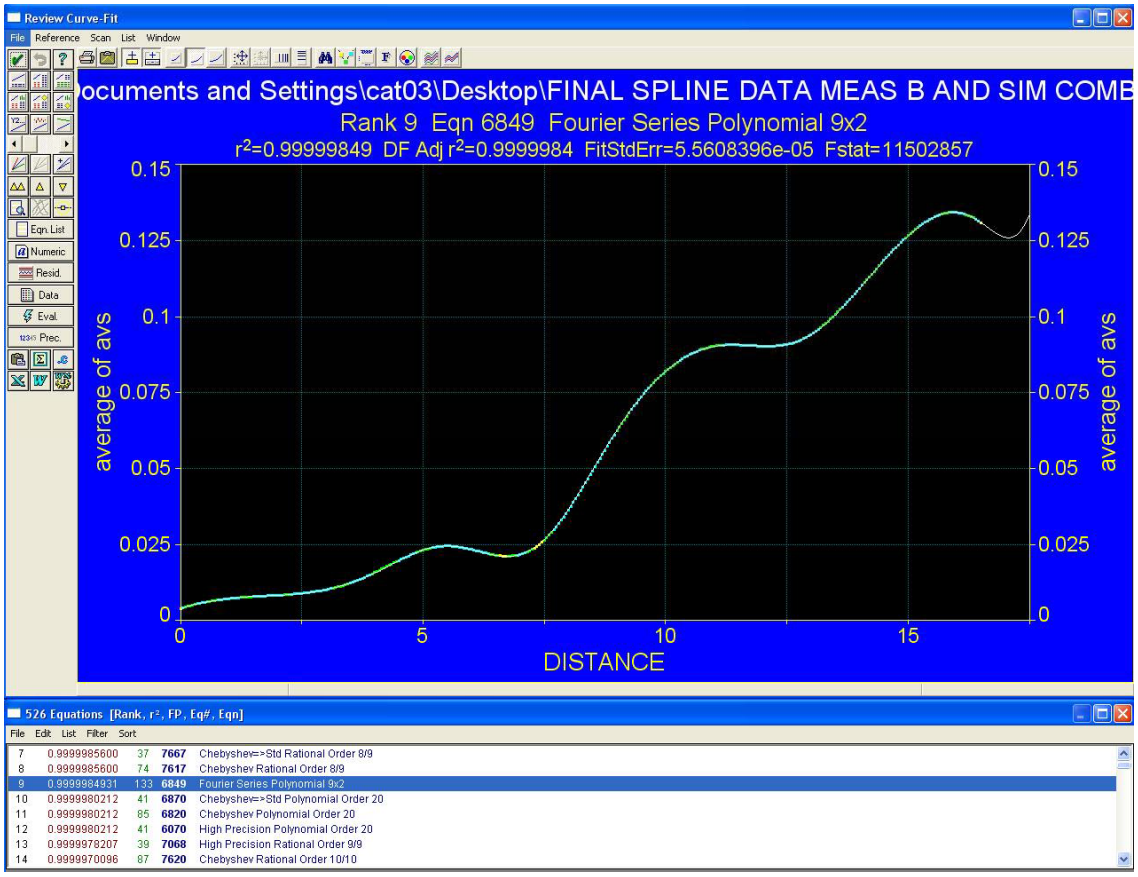

Figure 6.12. TableCurve 2D GUI showing the final spline approximation of the combined dataset of the experimental and simulator results.

The accuracy of the surgical simulator for cochlear implantation has been assessed using spline approximations of the output force data and by performing a statistical analysis of the results. Future developments and enhancements of the simulator can be validated using the same approach. Force profiles generated from

future insertion studies may be compared against the results and spline approximations obtained in this work. An increase in the dataset size, particularly the experimental results, will lead to a more thorough evaluation of the surgical simulator in comparison to real insertion data.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Overview

The primary focus of the thesis was to develop a haptic-rendered surgical simulator with the application to train specialists in CI insertion. The project work was directed at achieving this end goal and has been presented as a series of chapters in chronological order, corresponding to the successive stages of simulator development. The major aspects of the work included an analysis of the insertion process and force measurement, ST geometric model construction, visual and haptic rendering of virtual CI insertions and system validation.

The unique contributions of the work included a thorough insertion force study for implantation of the Contour and Contour Advance electrodes, derivation of a three-dimensional surface reconstruction of the human ST from measured data and visualisation of the structure in the Reachin API. Visual and haptic rendering of a virtual CI insertion was effectively implemented for interactive implantation of the sub-sampled Contour electrode into the ST model using the SIT. The final system was validated by performing statistical and qualitative comparisons between the output force profiles generated from the experimental work and simulated cochlear implantations. The simulator results are encouraging and reveal the benefits of a low-cost, risk-free virtual medical training facility, with the functionality to replicate patient-specific models for CI insertions. The system is the first of its kind to offer force feedback

during a virtual CI insertion. The novel, methodical approaches developed in this work may also be applied to other areas of surgical training.

7.2 Benefits of the Work

In the first chapter of the thesis, the overall concept of the project was introduced with an emphasis on the unique contributions of the work. The structure and mechanics of the human cochlea, and importantly, the process of cochlear implantation were reviewed to identify the principal structures of interest and design criteria for the work. An assessment of existing insertion studies and surgeon training schemes showed the importance of the work. The considerable benefits that would be offered by a low-cost, hazard-free surgical simulator of this type were proposed.

The overall objectives of the work were disclosed at the end of Chapter 1. A clinically valid surgical simulator was to be designed, constructed and validated for the purpose of supplementing current training schemes for CI insertions. Visual and force feedback were to be provided to the user during interactive, real-time device implantation into a three-dimensional virtual model of the human ST. The physical models of the ST and electrode were to use real-world data and their interactions during a virtual insertion would be compared with experimental results. Simulator validation was a necessary proviso of the design criteria, to quantify the precision of the simulator in mimicking a practical insertion and hence to evaluate the accuracy of the work.

7.3 Evolution in Surgical Simulation

In Chapter 2, a thorough background study of the work relating to visual and haptic rendering for surgical simulations was provided. This review disclosed the potential benefits that a surgical simulator for cochlear implantation with haptic feedback would

provide to the surgeon and research community, and revealed that a simulator of this type had not previously been developed. The work focused on visualisation techniques for rendering temporal bone anatomy, including the cochlea, as well as reviewing haptic rendering methods and applications in the area of medical education. The concept of haptic rendering was discussed, with an emphasis on real-time design considerations. Three main areas of focus for haptic rendering in surgical applications were identified to be: soft tissue manipulation, temporal bone drilling and device insertions. By reviewing existing visualisation and force-rendering applications, it was confirmed that the vast spectrum of haptic-rendering applications failed to include CI surgery and as such, the project work is a significant and unique contribution to the research. Through extensive investigation it was revealed that a surgical simulator which replicates the real-time, physical behaviour of an electrode array during its insertion into a model of the human ST had not previously been implemented. Haptic rendering of a CI process was unprecedented and possible reasons included historical limitations in technology, the complexity and scale of the structures and process, the distinctiveness of the procedure and real-time haptic loop constraints.

7.4 Cochlear Implant Insertion Studies

Insertion studies were performed to evaluate force delivery and implant behaviour during advancement of the Contour, Practice and Contour Advance electrodes into a two-dimensional, synthetic model of the human ST. The SIT and partial withdrawal methods were applied for the Contour and Practice, while the Contour Advance was inserted using the AOS technique. The method and results from the experimentation were discussed in Chapter 3. Frictional force experiments were also carried out to determine the coefficient of friction for the ST and electrode interface, for parameter

inclusion in the simulator. The insertion studies provided a comprehensive analysis of the insertion process for the purpose of its virtual replication, as well as yielding quantifiable results for simulator validation. The related literature did not encompass the insertion trials performed in the work and the results produced in Chapter 3 therefore extend the existing research. Key factors that were identified as contributing to force output during a CI insertion included: carrier strength, contact pressure, frictional force, electrode trajectory and surgical technique. From the insertion trials, the Contour array was selected as the electrode design to model, using the SIT, as a basis for simulator development.

7.5 Geometric Model of the Human ST

A three-dimensional, geometric model of the human ST (the primary passage for electrode insertion) was created for use in the simulation. This was the first stage in simulator construction and the content of Chapter 4. The literature relating to three-dimensional reconstructions of the cochlea from CT, MRI, histology, measured data and mathematical descriptions were assessed. A surface-based model of the cochlea was first derived from spiral CT, however the final rendering produced in Analyze was not suitable for this application since the cochlear spiral was inaccurately represented as a single cavity. A second, better approximation of the ST chamber was derived from measured data and the model was parameterised to make it patient-specific. Measurements that were used to form the ST model included: ST width, height and cross-sectional area, ST IW and OW measurements at each ¼ turn, OC length, radial lengths at each ¼ turn about the modiolar axis, cochlear axial height, cochlear diameters at each ½ turn and BM width variation. The ST polygonal surface model was realised in ANSYS. For replication, the model requires the parameters: OC length, ST cross-

section width and height at the site of the RW. Surface models of the Contour electrode and a flat approximation of the BM, as well as surgical tweezers, were also produced in ANSYS for use in the simulation.

ST model accuracy was confirmed by comparing the spiral with two mathematical descriptions of the ST. The cross-sectional areas of the ST were also calculated and compared with measured data. For the application of CI insertion, a surface description of the ST was deemed sufficient since the ST geometry does not change form during the insertion process. Only the carrier will flex during contact with the ST walls. The surface-based description of the ST proved effective as it complied with real-time constraints and increased computation processing speed for traversal of the haptic-rendering loop. The result of Chapter 4 was an anatomically accurate, three-dimensional surface reconstruction of the human ST derived from measured data for a virtual CI insertion.

7.6 Real-time Visual and Haptic Rendering of CI Insertion

Simulator design and implementation were presented in Chapter 5. The focus of the work in this chapter was on the visual and haptic rendering of the CI insertion, for modelling the SIT with the Contour electrode. The supporting infrastructure, including the Reachin API software and haptic-rendering hardware, was discussed, as well as the benefits associated with using this system. Surface models of the ST, BM, implant, stylet and surgical tweezers produced in ANSYS, were visualised in the Reachin API. System functionality included scene magnification, object rotation and translation, selection of the carrier using the haptic device button and interactive carrier mobility. Frictional surface properties were added to the ST for haptic-rendering, however due to stability issues, program optimisations were required. Significant reductions in file sizes

and polygon count for surface representations enabled real-time haptic rendering. Carrier sub-sampling was further required to produce viable visual and haptic rendering results, to satisfy real-time constraints. Full insertion of the carrier was realised using real-time re-sampling of the carrier based on progressive electrode insertion depth. A force vector was added to the model to account for frictional force accumulation during electrode advancement. During ST and electrode interactions, output force and position data were logged to a text file as well as being displayed on the GUI. Physical properties of the ST and carrier based on measured data, such as coefficients of friction, were included in the simulation to make it more realistic. During program execution, the user may manipulate the virtual environment with the haptics device. As the carrier is inserted into the three-dimensional surface model of the human ST, real-time force and torque feedback are provided to the user, synthesising the virtual cochlear implantation of a sub-sampled Contour array using the SIT to fully insert the implant.

7.7 System Validation

Simulator accuracy was evaluated in Chapter 6 by comparing force profiles produced from the simulation with results obtained from the insertion force experiments from Chapter 3. Spline approximations of the results obtained from the simulation were created using TableCurve 2D and processed to produce an average curve of best fit for the insertion force data. This result was compared with an average spline approximation produced in a similar way for the experimental results. A statistical comparison revealed that the difference between the average curves varied about 0N, with a maximum difference of magnitude 0.0536N. By considering the differences and the 95% confidence intervals for the entire electrode displacement, regions were assessed in terms of statistical significance and practical importance. It was concluded that the

differences between the experimental and simulator results were minimal and for regions of statistical significance, the differences were not of practical importance. The simulator was validated by performing this statistical and qualitative comparison between the practical and simulation results. As a final stage in the work, the insertion force profiles that were produced from the insertion force experiments and the simulations were combined into one spline approximation in order to represent the average output force for any given insertion as a function of implant displacement along the ST. The final curve and equation describing the spline can be used for validation of future insertion results.

The project work presented throughout the chapters of the dissertation addresses the overall objectives and challenges of the research. A haptic-rendered surgical simulator was designed, constructed and validated with experimental data, for the purpose of training surgeons to perform CI insertions in a safe, reproducible and commercially viable environment. Visual and force feedback were provided to the user during a virtual, real-time insertion where the user interactively advanced a sub-sampled model of the Contour array into a three-dimensional, anatomically accurate model of the human ST, using the SIT. Physical properties of the ST and carrier were replicated from real-world data. The results of the simulation were qualitatively and statistically compared with practical insertion results, which revealed the accuracy of the simulation, particularly in the region of the Basal turn.

7.8 Major Contributions of the Work

In summary, the benefits of this type of surgical simulator include the provision of a safe, cost-effective, reliable and reproducible training environment in which the patient is not at risk of induced trauma. A patient-specific model captures individual variations

in cochlear shape, which is useful for pre- and post- operative planning. The simulator offers the unique functionality for real-time force measurement and feedback to the user, as well as providing real-time insertion depth information, which has previously been subjective. This type of information can assist surgeons in their administration techniques, as well as provide useful information to the implant manufacturer for future design enhancements.

The simulator is the first of its kind to provide real-time force feedback and offer an objective evaluation of force delivery by quantifying insertion forces, for the maximally invasive operation of cochlear implantation. Since information is inferred tactually for this type of surgery, the solution presented in the thesis offers a new platform for improving dexterity in this area. It is envisaged that the method is extensible to other areas of haptic skill acquisition, such as biopsies or alternative prosthetic implantations. The unique approaches and results presented in the thesis for the production of a haptic-rendered surgical simulator that enables real-time cochlear implantations into a virtual, accurate model of the human ST reveals the significant contributions of the work.

7.9 Current Constraints and Future Work

The major constraints of the developed system are imposed by the collision response algorithm which determines the sub-sampled form of the electrode carrier during its interaction with the ST walls. Improvements in the algorithm are required to properly model the changing form of the carrier and the positions of the sub-samples comprising its structure, as it progresses along the spiralling walls of the ST, particularly after the region of the Basal turn. In relation to the insertion force experimentation, there are some constraints associated with the synthetic model of the ST which have been used to

validate the simulation. The simulator should also be subjectively reviewed by an experienced otologist that performs CI surgery.

7.9.1 Contour Collision Response Algorithm

The model of the sub-sampled carrier in the simulation is an approximation of the Contour electrode and during its insertion, the physical flexing of the model is not ideally represented following its interaction with the ST walls. The algorithm that implements this complex nature of collision response, particularly after the Basal turn region, requires improvement. This would involve reviewing the process for real-time sub-sampling of the carrier and repositioning the sub-samples following their interactions with the ST walls. The algorithm which implements the collision response of the carrier and its changing of form provides a template for this development. The challenges involved in the visual and haptic rendering of the carrier during collision response include: determining the new position of the sample point following its collision with the ST walls, the complex shape and variability of the human cochlea model, as well as haptic loop update rate constraints. The latter restricts the complexity of the haptic algorithms as well as the physical representations of the structures and is a primary concern for all haptic-rendered simulator developers. Improvements in the carrier response and its trajectory would increase the accuracy of the simulator. This may reduce the difference in average output force, as detailed in Chapter 6.

7.9.2 Insertion Force Measurement

There are a number of improvements that could be made in order to improve the insertion force results obtained via experimentation. A two-dimensional Teflon model of the human ST is used in the experimentation; however it lacks the internal tissue

composition and three-dimensional shape of a live human cochlea. Ideally, force measurements should be performed in vivo or the current results compared with in vitro insertions into human cadavers. A greater number of insertion trials need to be performed via experimentation, in order to further substantiate both the output force profiles recorded and the simulator accuracy. A 1DOF Instron force measurement device is used to record output force along one direction only in the experimentation and other force components (in the perpendicular planes), as well as torques, which are currently neglected should be measured. The physical, real-world data included in the virtual model of the ST represent the properties of the synthetic replica, which is of different material composition to the true human cochlea.

Ideally, in order to validate the simulation, the experimental tests should be performed in vivo. The current technology does not yet allow for accurate measurement of insertion force data in a live patient and so insertion force studies have historically used synthetic models of the ST. A three-dimensional synthetic model is preferred to a two-dimensional model, as the electrode trajectory is more precise. However, it is hard to replicate the tissue properties and measure insertion forces that induce trauma in a plastic model. In future studies, a fresh cochlear cadaver may be used, where its tissue composition and three-dimensional shape more closely resemble the true scenario. This will require ethics approval, material acquisition and issues concerning storage should also be considered. The results produced from insertion analyses using cadaver material may be compared with the results obtained in this work. Future work should examine the limitations in the current force measurement technology (including size and intrusiveness during device insertions into small, delicate structures such as the cochlea). Overcoming these challenges might involve hardware developments or modifications that enable in vivo force measurement during a cochlear implantation. However,

liability and feasibility issues of inserting additional hardware or monitoring devices into the cochlear during a live implantation should be carefully and cautiously observed.

A greater number of insertion trials should be carried out, since experimental results were obtained for only 11 tests compared to 200 simulation trials. This would strengthen the conviction of the results presented in Chapter 3 and further substantiate the simulator accuracy evaluated in Chapter 6. The spline approximations and associated equations that represent the experimental results as well as the combined results for the simulator and experimental work, as detailed in Chapter 6, may be used for comparison of new results. The simulation should be separately evaluated by an experienced ENT surgeon in order to identify any changes or enhancements that are required. This should be done to ensure that the simulator is suitable for medical education purposes, specifically for training surgeons in CI insertions.

A 1DOF Instron device is used in the insertion force experimentation. This does not measure all output force components but only the net force along the longitudinal axis of the ST. Ideally, a 6DOF instrument should be used, so that all force components are considered and the results can be compared with the equivalent output forces and torques from the simulation. Further, the sampling frequency in the experimentation should match that used in the simulation.

The physical properties of the ST model which are included in the simulation represent values obtained from the synthetic environment. Whilst it is consistent with the physical environment of the experiments, it may deviate from the physical structure of a live human cochlea. However, physical properties such as the coefficients of friction between an electrode and a live human cochlea have not been published. Physical properties such as these are not easily measurable and therefore the synthetic approximation is often used.

7.9.3 Virtual Cochlear Model

The ST model used in the simulation is based on measured data and although it is parameterised, the data must be extracted from CT images to make it patient-specific. Whilst CT images are taken pre-operatively for cochlear implantations, the problem of exact feature extraction has not been addressed and there may be variability in the method and accuracy of the measurements. Further, anomalies in cochlear conditions such as ossification or dysplasia are not included in the surface model. If the model is derived from CT, there is more likelihood of capturing these anomalies as well as enabling semi-automatic cochlear extraction and surface rendering of the structure, so that inaccuracies in measurement are expelled. Refinement in the process of extracting the cochlea from CT that was described in Chapter 3, so that the ST remains a single cavity, is recommended and future work may include multi-modal image registration of CT with MRI to improve structure delineation.

Cochlear structures other than the ST, such as the CP (including the BM) and SV, may also be included in the simulation as well as surrounding temporal bone anatomy. This would enhance the virtual experience for the training surgeon and improve skills in landmark identification. A flat approximation of the BM, produced in the work, should be included in the simulation as a membranous surface. The current model enables addition of the CP and SV, which may be constructed from measured data using a similar approach for object visualisation and haptic rendering as described in the work. It may be of interest to add the SV as an alternative passage for CI insertion. Inclusion of the mastoid quadrant of the temporal bone in the simulation for temporal bone dissection may lead to the variation of the cochleostomy site, which the user could interactively approach and expose via bone drilling. Currently, the cochleostomy site is fixed and the angle of approach is arbitrary about this opening.

7.9.4 Stylet Withdrawal in the Simulation

The stylet and its withdrawal have not been included in the current simulation. For a practical CI insertion using the SIT, the stylet is withdrawn after the electrode is fully inserted into the ST. The functionality for a stylet withdrawal has been implemented but was not included in the final simulation, due to the requirement for reductions in simulator complexity. This may be re-introduced and future developments should include the modelling of the recoil properties of the perimodiolar electrode design as the stylet is progressively withdrawn, at the end of the insertion. The current simulation does not enable stylet withdrawal that results in the carrier curling towards the inner ST wall. Advancements in collision response for the carrier should address this issue. Once the functionality is added, the insertion techniques for the partial withdrawal method and AOS insertion, with a new model for the Contour Advance electrode, may also be introduced into the simulation. This will provide the surgeon with the option of administration technique for the two types of implants manufactured by Cochlear^{TM}, that are available to surgeons world-wide.

During a live implantation, the surgeon uses the dominant hand to hold the tweezers and insert the implant. In the simulation, the haptics device represents the surgical tool that grasps the virtual implant. Only one haptic device is used in the simulation for the implantation of the carrier, whereas the surgeon will also use the contralateral hand to perform tasks such as stylet withdrawal during a live implantation. In this application, a second haptics device should be included, so that the surgeon can make a selection from a range of tools to perform other tasks during the insertion. A surgical hook is commonly used to withdraw the stylet with the contralateral hand. For this purpose, a 3DOF haptics device would be sufficient.

7.9.5 Surgeon Evaluation

The simulator has been objectively validated by performing a statistical comparison of force profiles, between experimental and simulated insertion results. Live insertions were also examined and insertion studies were analysed during the design and implementation stages. However, subjective evaluation by an experienced ENT surgeon that performs CI surgery is desirable. The intent of the work is to include the simulator in medical education programs and as such, medical specialists should evaluate the CI simulator and offer suggestions to improve the simulator for educational purposes. There may be specific functions that are desirable for medical teaching and learning, such as interactive landmark delineation, visual fly-throughs of the anatomy within this region or an inbuilt assessment system for evaluation of surgeon skill.

References

[1] J. M. Palmer, "The Ear," in *Anatomy for Speech and Hearing*, 1993, pp. 175-250.

[2] R. A. Houde, J. M. Pickett and H. Levitt, "Prologue: Auditory and Speech Impairments," in *Sensory Training Aid for the Hearing Impaired*, H. Levitt, J. M. Pickett, and R. A. Houde, Eds., 1980, pp. 3-27.

[3] M. W. Skinner, D. R. Ketten, L. K. Holden, *et al.*, "CT-Derived Estimation of Cochlear Morphology and Electrode Array Position in Relation to Word Recognition in Nucleus-22 Recipients," *Journal of the Association for Research in Otolaryngology*, vol. 03, pp. 332-350, 2002.

[4] I. K. Indrajit, J. D. Souza, V. K. Singh, *et al.*, "Imaging of Cochlear Implants," *Indian Journal of Radiology and Imaging*, vol. 13, pp. 371-379, 2003.

[5] P. Hans, A. J. Grant, R. D. Laitt, *et al.*, "Comparison of Three-Dimensional Visualization Techniques for Depicting the Scala Vestibuli and Scala Tympani of the Cochlea by Using High-Resolution MR Imaging," *The American Journal of Neuroradiology*, vol. 20, pp. 1197-1206, 1999.

[6] J. H. M. Frijns, J. J. Briaire and J. J. Grote, "The Importance of Human Cochlear Anatomy for the Results of Modiolus-Hugging Multichannel Cochlear Implants," *Otology & Neurotology*, vol. 22, pp. 340-349, 2001.

[7] S. K. Yoo, G. Wang, J. T. Rubinstein, *et al.*, "Three-Dimensional Geometric Modeling of the Cochlea Using Helico-Spiral Approximation," *IEEE Transactions on Biomedical Engineering*, vol. 47, pp. 1392-1402, 2000.

[8] D. R. Ketten, M. W. Skinner, G. Wang, *et al.*, "In Vivo Measures of Cochlear Length and Insertion Depth of Nucleus Cochlear Implant Electrode Arrays," *Annals of Otology, Rhinology & Laryngology*, vol. 107, pp. 1-16, 1998.

[9] A. Kawano, H. L. Seldon and G. M. Clark, "Computer-Aided Three-Dimensional Reconstruction in Human Cochlear Maps: Measurement of the Lengths of Organ of Cort, Outer Wall, Inner Wall, and Rosenthal's Canal," *Annals of Otology, Rhinology & Laryngology*, vol. 105, pp. 701-709, 1996.

[10] A. Takagi and I. Sando, "Computer-Aided Three-Dimensional Reconstruction: A Method of Measuring Temporal Bone Structures including the Length of the Cochlea," *Annals of Otology, Rhinology & Laryngology*, vol. 98, pp. 515-522, 1989.

[11] L. T. Cohen, J. Xu, S. A. Xu, *et al.*, "Improved and Simplified Methods for Specifying Positions of the Electrode Bands of a Cochlear Implant Array," *The American Journal of Otology*, vol. 17, pp. 859-865, 1996.

[12] Y. Raphael and R. A. Altschuler, "Structure and Innervation of the Cochlea," *Brain Research Bulletin*, vol. 60, pp. 397-422, 2003.

[13] H. E. Davis, "Hearing and Deafness." New York, Holt, Rinehart and Winston, Inc., 1961.

[14] P. C. Loizou, "Introduction to Cochlear Implants," *IEEE Engineering in Medicine and Biology*, vol. 18, pp. 32-42, 1999.

[15] J. J. May, "Occupational Hearing Loss," *American Journal of Industrial Medicine*, vol. 37, pp. 112-120, 2000.

[16] G. T. Porter, "Cochlear Implants," Grand Rounds Presentation, UTMB, Dept. of Otolaryngology, F. B. Quinn Jr. and M. W. Ryan, Eds., 2003.

[17] AIHW, "Annual Report 2004/2005 to Parliament," Australian Government; Department of Health and Aging, 2005.

[18] AIHW, "Australia's Health 2002," The Australian Institute of Health and Welfare, biennial report to Parliament, 2002.

[19] J. Kronenberg and L. Migirov, "The Role of Mastoidectomy in Cochlear Implant Surgery," *Acta Otolaryngology*, vol. 123, pp. 219-222, 2003.

[20] J. A. Gulya and R. L. Steenerson, "The Scala Vestibuli for Cochlear Implantation," *Otolaryngology - Head and Neck Surgery*, vol. 122, pp. 130-133, 1996.

[21] E. Pasanisi, V. Vincenti, A. Bacciu*, et al.*, "The Nucleus Contour Electrode Array: An Electrophysiological Study," *The Laryngoscope*, vol. 112, pp. 1653-1656, 2002.

[22] K. R. Stidham and J. Roberson Jr., B., "Cochlear Hook Anatomy: Evaluation of the Spatial Relationship of the Basal Cochlear Duct to Middle Ear Landmarks," *Acta Otolaryngology*, vol. 119, pp. 773-777, 1999.

[23] J. T. J. Roland, "Cochlear Implant Electrode Insertion," *Operative Techniques in Otolaryngology-Head and Neck Surgery*, vol. 16, pp. 86-92, 2005.

[24] Cochlear™, "Cochlear Timeline: A History of Innovation," Cochlear™, Ed., sited 30/11/2005, http://www.cochlear.com/116.asp.

[25] N. L. Cohen, J. T. Roland, Jr. and A. Fishman, "Surgical Technique For The Nucleus Contour Cochlear Implant," *Ear & Hearing*, vol. 23, pp. 59S-66S, 2002.

[26] B. K. Chen, G. M. Clark and R. Jones, "Evaluation of Trajectories and Contact Pressures for the Straight Nucleus Cochlear Implant Electrode Array - a Two-Dimensional Application of Finite Element Analysis," *Medical Engineering & Physics*, vol. 25, pp. 141-147, 2003.

[27] M. A. Marsh, J. Xu, P. J. Blamey*, et al.*, "Radiologic Evaluation of Multichannel Intracochlea Implant Insertion Depth," *The American Journal of Otology*, vol. 14, pp. 386-391, 1993.

[28] C.-H. Chang, G. T. Anderson and P. C. Loizou, "A Neural Network Model for Optimizing Vowel Recognition by Cochlear Implant Listeners," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, pp. 42-48, 2001.

[29] J. B. Hinderink, L. H. Mens, J. P. Brokx*, et al.*, "Performance of Prelingually and Postlingually Deaf Patients using Single-Channel or Multi-Channel Cochlear Implants," *Laryngoscope*, vol. 105, pp. 618-622, 1995.

[30] T. Lenarz, J. Kuzma, B. P. Weber*, et al.*, "New Clarion Electrode with Positioner: Insertion Studies," presented at Seventh Symposium on Cochlear Implants in Children, Iowa City, Iowa, pp. 16-18, 1998.

[31] M. Tykocinski, E. Saunders, L. T. Cohen*, et al.*, "The Contour Electrode Array: Safety Study and Initial Patient Trials of a New Perimodiolar Design," *Otology & Neurotology*, vol. 22, pp. 33-41, 2001.

[32] E. Saunders, L. Cohen, A. Aschendorff*, et al.*, "Threshold, Comfortable Level and Impedance Changes as a Fuction of Electrode-Modiolar Distance," *Ear & Hearing*, vol. 23, pp. 28S-40S, 2002.

[33] W. Gstoettner, P. Franz, J. Hamzavi, *et al.*, "Intracochlear Position of Cochlear Implant Electrodes," *Acta Otolaryngology*, vol. 119, pp. 229-233, 1999.

[34] W. K. Gstoettner, O. Adunka, P. Franz, *et al.*, "Perimodiolar Electrodes in Cochlear Implant Surgery," *Acta Otolaryngology*, vol. 121, pp. 216-219, 2001.

[35] S. J. Rebscher, M. Heilmann, W. Bruszewski, *et al.*, "Strategies to Improve Electrode Positioning and Safety in Cochlear Implants," *IEEE Transactions on Biomedical Engineering*, vol. 46, pp. 340-352, 1999.

[36] S. J. Rebscher, N. Talbot, W. Bruszewski, *et al.*, "A Transparent Model of the Human Scala Tympani Cavity," *Journal of Neuroscience Methods*, vol. 64, pp. 105-114, 1996.

[37] R. K. Shepherd, G. M. Clark, B. C. Pyman, *et al.*, "Banded Intracochlear Electrode Array: Evaluation of Insertion Trauma in Human Temporal Bones," *Annals of Otology, Rhinology & Laryngology*, vol. 94, pp. 55-59, 1985.

[38] B. Richter, A. Aschendorff, P. Lohnstein, *et al.*, "The Nucleus Contour Electrode Array: A Radiological and Histological Study," *The Laryngoscope*, vol. 111, pp. 508-514, 2001.

[39] D. W. Kennedy, "Multichannel Intracochlear Electrodes: Mechanism of Insertion Trauma," *Laryngoscope*, vol. 97, pp. 42-49, 1987.

[40] P. Wardrop, D. Whinney, S. J. Rebscher, *et al.*, "A Temporal Bone Study of Insertion Trauma and Intracochlear Position of Cochlear Implant Electrodes. I: Comparison of Nucleus Banded and Nucleus Contour™ Electrodes," *Hearing Research*, vol. 203, pp. 54-67, 2005.

[41] Y. S. Lim, S.-I. Park, Y. H. Kim, *et al.*, "Three-Dimensional Analysis of Electrode Behavior in a Human Cochlear Model," *Medical Engineering & Physics*, vol. 27, pp. 695-703, 2005.

[42] H. N. Kha, B. K. Chen, G. M. Clark, *et al.*, "Stiffness properties for Nucleus standard straight and contour electrode arrays," *Medical Engineering & Physics*, vol. 26, pp. 677-685, 2004.

[43] A. A. Eshraghi, N. W. Yang and T. J. Balkany, "Comparative Study of Cochlear Damage With Three Perimodiolar Electrode Designs," *The Laryngoscope*, vol. 113, pp. 415-419, 2003.

[44] D. B. Welling, R. Hinojosa, B. J. Gantz, *et al.*, "Insertional Trauma of Multichannel Cochlear Implants," *Laryngoscope*, vol. 103, pp. 995-1001, 1993.

[45] N. W. Yang, A. V. Hodges and T. J. Balkany, "Novel Intracochlear Electrode Positioner: Effects on Electrode Position," presented at Seventh Symposium on Cochlear Implants in Children, Iowa City, Iowa, pp. 18-20, 1998.

[46] O. Adunka, J. Kiefer, M. H. Unkelbach, *et al.*, "Development and Evaluation of an Improved Cochlear Implant Electrode Design for electric Acoustic Stimulation," *The Laryngoscope*, vol. 114, pp. 1237-1241, 2004.

[47] J. B. Nadol Jr., J. Y. Shiao, B. J. Burgess, *et al.*, "Histopathology of Cochlear Implants in Humans," *The Annals of Otology, Rhinology & Laryngology*, vol. 110, pp. 883-891, 2001.

[48] P. R. Deman, K. Daemers, M. Yperman, *et al.*, "Basic Fitting and Evaluation Parameters of a Newly Designed Cochlear Implant Electrode," *Acta Otolaryngology*, vol. 124, pp. 281-285, 2004.

[49] P. A. Wackym, J. B. Firszt, W. Gaggl, *et al.*, "Electrophysiologic Effects of Placing Cochlear Implant Electrodes in a Perimodiolar Position in Young Children," *The Laryngoscope*, vol. 114, pp. 71-76, 2004.

[50] T. P. Mason, E. L. Applebaum, M. Rasmussen, *et al.*, "Virtual temporal bone: Creation and application of a new computer-based teaching tool," *Otolaryngology - Head and Neck Surgery*, vol. 122, pp. 168-173, 2000.

[51] C. Jones, Q. Sun and R. Z. Gan, "Computer-Aided 3-Dimensional Modeling of Human Ear," presented at Proceedings of the Second Joint EMBS/BMES Conference, Houston, TX, USA, pp. 266, 2002.

[52] J. Xu, S.-A. Xu, L. T. Cohen, *et al.*, "Cochlear View: Postoperative Radiography for Cochlear Implantation," *The American Journal of Otology*, vol. 21, pp. 49-56, 2000.

[53] H. Takahashi and I. Sando, "Computer-Aided 3-D Temporal Bone Anatomy for Cochlear Implant Surgery," *Laryngoscope*, vol. 100, pp. 417-421, 1990.

[54] R. Z. Gan and Q. Sun, "Finite Element Modeling of Human Ear with External Ear Canal and Middle Ear Cavity," presented at Proceedings of the Second Join EMBS/BMES Conference, Houston, TX, USA, pp. 264-265, 2002.

[55] D. R. Ketten and D. Wartzok, "Three-Dimensional Reconstructions of the Dolphin Ear," in *Sensory Abilities of Cetaceans*. New York, Plenum Press, 1990.

[56] A. H. Voie and F. A. Spelman, "Analysis of the Guinea Pig Cochlea using a General Cylindrical Coordinate System," presented at Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 0206, 1990.

[57] C. Lutz, A. Takagi, I. P. Janecka, *et al.*, "Three-dimensional Computer Reconstruction of a Temporal Bone," *Otolaryngology - Head and Neck Surgery*, vol. 101, pp. 522-526, 1989.

[58] T. Harada, S. Ishii and N. Tayama, "Three-dimensional Reconstruction of the Temporal Bone from Histologic Sections," *Otolaryngology - Head and Neck Surgery*, vol. 114, pp. 1139-1142, 1988.

[59] Y. Nomura, T. Okuno, M. Hara, *et al.*, "Walking through a Human Ear," *Acta Otolaryngology*, vol. 107, pp. 366-370, 1989.

[60] D. J. Green, M. S. Marion, B. J. Erickson, *et al.*, "Three-Dimensional Reconstruction of the Temporal Bone," *Laryngoscope*, vol. 100, pp. 1-4, 1990.

[61] S. J. Daniel, W. R. J. Funnell, A. G. Zeitouni, *et al.*, "Clinical Application of a Finite-Element Model of the Human Middle Ear," *The Journal of Otolaryngology*, vol. 30, pp. 340-346, 2001.

[62] P. A. Warrick and R. J. Funnell, "A VRML-Based Anatomical Visualization Tool for Medical Education," *IEEE Transactions on Information Technology in Biomedicine*, vol. 2, pp. 55-61, 1998.

[63] R. G. Van Wijhe, "A Finite-Element Model of the Middle Ear of the Moustached Bat." Masters of Engineering, Montreal, McGill University, 2000.

[64] A. H. Voie, F. A. Spelman, D. Sutton, *et al.*, "Quantitative Measurements of the Three-Dimensional Anatomy of the Guinea Pig Cochlea Using Optical Sectioning," presented at IEEE Engineering in Medicine & Biology Society 11th Annual International Conference, pp. 1765-1766, 1989.

[65] T. P. Mason, E. L. Applebaum, M. Rasmussen, *et al.*, "The Virtual Temporal Bone," presented at Medicine Meets Virtual Reality, pp. 346-352, 1998.

[66] G. J. Wiet, D. Stredney, D. Sessanna, *et al.*, "Virtual Temporal Bone Dissection: An Interactive Surgical Simulator," *Otolaryngology - Head and Neck Surgery*, vol. 127, pp. 79-83, 2002.

[67] M. G. Qiu, S. X. Zhang, Z. J. Liu, *et al.*, "Visualization of the Temporal Bone of the Chinese Visible Human," *Surgical & Radiologic Anatomy*, vol. 26, pp. 149-152, 2004.

[68] S. Boor, J. Maurer, W. Mann, *et al.*, "Virtual Endoscopy of the Inner Ear and the Auditory Canal," *Head and Neck Radiology*, vol. 42, pp. 543-547, 2000.

[69] S. K. Yoo, G. Wang, J. T. Rubinstein, *et al.*, "Three-Dimensional Modeling and Visualization of the Cochlea on the Internet," *IEEE Transactions on Information Technology in Biomedicine*, vol. 4, pp. 144-151, 2000.

[70] M. D. Seemann, O. Seemann, H. Bonel, *et al.*, "Evaluation of the middle and inner ear structures: comparison of hybrid rendering, virtual endoscopy and axial 2D source images," *European Radiology*, vol. 9, pp. 1851-1858, 1999.

[71] T. Rodt, P. Ratiu, H. Becker, *et al.*, "3D Visualisation of the Middle Ear and Adjacent Structures using Reconstructed Multi-Slice CT Datasets, Correlating 3D Images and Virtual Endoscopy to the 2D Cross-Sectional Images," *Head and Neck Radiology*, vol. 44, pp. 783-790, 2002.

[72] T. Himi, A. Kataura, M. Sakata, *et al.*, "Three-Dimensional Imaging of the Temporal Bone Using a Helical CT Scan and Its Application in Patients with Cochlear Implantation," *Otorhinolaryngology*, vol. 58, pp. 298-300, 1996.

[73] M. W. Skinner, D. R. Ketten, M. W. Vannier, *et al.*, "Determination of the Position of Nucleus Cochlear Implant Electrodes in the Inner Ear," *The American Journal of Otology*, vol. 15, pp. 644-651, 1994.

[74] B. Pflesser, A. Petersik, U. Tiede, *et al.*, "Volume Cutting for Virtual Petrous Bone Surgery," *Computer Aided Surgery*, vol. 7, pp. 74-83, 2002.

[75] W. F. Decraemer, J. J. J. Dirckx and W. R. J. Funnell, "Three-Dimensional Modelling of the Middle-Ear Ossicular Chain Using a Commercial High-Resolution X-Ray CT Scanner," *Journal of the Association for Research in Otolaryngology*, vol. 04, pp. 250-263, 2003.

[76] H. L. Seldon, "Three-Dimensional Reconstruction of Temporal Bone from Computed Tomographic Scans on a Personal Computer," *Otolaryngology - Head and Neck Surgery*, vol. 117, pp. 1158-1161, 1991.

[77] F. Spoor and F. Zonneveld, "Comparative Review of the Human Bony Labyrinth," in *Yearbook of Physical Anthropology*, vol. 41, Wiley-Liss Inc., 1998, pp. 211-251.

[78] B. Pflesser, U. Tiede, K. H. Hohne, *et al.*, "Volume Based Planning and Rehearsal of Surgical Interventions," presented at Computer Assisted Radiology and Surgery, Proc. CARS 2000, Amsterdam, pp. 607-612, 2000.

[79] S. K. Yoo, G. Wang, F. Collison, *et al.*, "Three-Dimensional Localization of Cochlear Implant Electrodes Using Epipolar Stereophotogrammetry," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 838-846, 2004.

[80] G. Wang, M. W. Vannier, M. W. Skinner, *et al.*, "Spiral CT Image Deblurring for Cochlear Implantation," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 251-262, 1998.

[81] S. H. Bartling, K. Peldschus, T. Rodt, *et al.*, "Registration and Fusion of CT and MRI of the Temporal Bone," *Computer Assisted Tomography*, vol. 29, pp. 305-10, 2005.

[82] P. Hans, A. Jackson, J. E. Gillespie, *et al.*, "Virtual Reality Modelling Language: Freely Available Cross-Platform Visualization Technique for 3-D Visualization of the Inner Ear," *The Journal of Laryngology and Otology*, vol. 117, pp. 766-774, 2003.

[83] M. Thorne, A. N. Salt, J. E. DeMott, *et al.*, "Cochlear Fluid Space Dimensions for Six Species Derived From Reconstructions of Three-Dimensional Magnetic Resonance Images," *Laryngoscope*, vol. 109, pp. 1661-1668, 1999.

[84] M. Duan, B. Bjelke, A. Fridberger, *et al.*, "Imaging of the Guinea Pig Cochlea following Round Window Gadolinium Application," *Neuroreport*, vol. 15, pp. 1927-1930, 2004.

[85] G. Wang, M. W. Vannier, M. W. Skinner, *et al.*, "Unwrapping Cochlear Implants by Spiral CT," *IEEE Transactions on Biomedical Engineering*, vol. 43, pp. 891-899, 1996.

[86] A. Aschendorff, R. Kubalek, A. Hochmuth, *et al.*, "Imaging Procedures in Cochlear Implant Patients: Evaluation of Different Radiological Techniques," *Acta Oto-Laryngologica*, vol. 552, pp. 46-49, 2004.

[87] M. Jiang, G. Wang, M. W. Skinner, *et al.*, "Blind Deblurring of Spiral CT Images," *IEEE Transactions on Medical Imaging*, vol. 22, pp. 837-845, 2003.

[88] K. H. Höhne, M. Bomans, A. Pommert, *et al.*, "3D Visualization of Tomographic Volume Data using the Generalized Voxel Model," *The Visual Computer*, vol. 6, pp. 28-36, 1990.

[89] B. Lichtenbelt, R. Crane and S. Naqvi, "Introduction to Volume Rendering," Hewlett-Packard Company, 1998.

[90] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, vol. 21, pp. 163-169, 1987.

[91] E. Givelberg, J. Bunn and M. Rajan, "Detailed Simulation of the Cochlea: Recent Progress Using Large Shared Memory Parallel Computers," Technical Report, California Institute of Technology, Caltech CACR, 2001.

[92] E. Givelberg and J. Bunn, "Computational Experiments with a Three-Dimensional Model of the Cochlea," Technical Report, California Institute of Technology, Caltech CACR, 2004.

[93] A. Eiber, "Mechanical Modeling and Dynamical Behavior of the Human Middle Ear," *Audiology & Neuro Otology*, vol. 4, pp. 170-177, 1999.

[94] C. S. Brown, A. C. Nelson and F. A. Spelman, "Three-dimensional Reconstruction of the Cochlea," presented at IEEE Engineering in Medicine & Biology Society 11th Annual International Conference, pp. 0536-0537, 1989.

[95] T. Hanekom, "Modelling of the electrode-auditory nerve fibre interface in cochlear prostheses." PhD (Electronic Engineering), University of Pretoria, 2001.

[96] K. Lundin, A. Ynnerman and B. Gudmundsson, "Proxy-based Haptic Feedback from Volumetric Density Data," presented at Eurohaptics 2002, Edinburgh, pp. 104-109, 2002.

[97] A. Gregory, A. Mascarenhas, S. Ehmann, *et al.*, "Six Degree-of-Freedom Haptic Display of Polygonal Models," *IEEE Visualization*, pp. 139-146, 2000.

[98] O. U. Press, "Compact Oxford English Dictionary," Oxford University Press, 2005.

[99] E. Acosta, B. Stephens, B. Temkin, *et al.*, "Development of a Haptic Virtual Environment," presented at 12th IEEE Symposium on Computer-Based Medical Systems (CBMS '99), pp. 35-39, 1999.

[100] J. Kim, S. De and M. A. Srinivasan, "Computationally Efficient Techniques for Real Time Surgical Simulation with Force Feedback," presented at Proceedings of the 10th Symp. On Haptic Interfaces for Virtual Envir. & Teleoperator Systs. (Haptics '02), pp. 51-57, 2002.

[101] J. E. Colgate, M. C. Stanley and J. M. Brown, "Issues in the Haptic Display of Tool Use," presented at Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA, pp. 140-145, 1995.

[102] S. Cotin and H. Delingette, "Real-time Surgery Simulation with Haptic Feedback using Finite Elements," presented at Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, pp. 3739-3744, 1998.

[103] O. R. Astley and V. Hayward, "Design Constraints for Haptic Surgery Simulation," presented at Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, 2000.

[104] R. Technologies, "Reachin API 3.2 Programmer's Guide." Stockholm, 2002.

[105] R. Baumann and R. Clavel, "Haptic Interface for Virtual Reality Based Minimally Invasive Surgery Simulation," presented at Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pp. 381-386, 1998.

[106] K. H. Höhne, "Guest Editorial: Medical Image Computing at the Institute of Mathematics and Computer Science in Medicine, University Hospital Hamburg-Eppendorf," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 713-723, 2002.

[107] V. Hayward, O. R. Astley, M. Cruz-Hernandez*, et al.*, "Haptic Interfaces and Devices," *Sensor Review*, vol. 24, pp. 16-29, 2004.

[108] D. C. Ruspini, K. Kolarov and O. Khatib, "Haptic Interaction in Virtual Environments," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97), pp. 128-133, 1997.

[109] W. A. McNeely, K. D. Puterbaugh and J. J. Troy, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," presented at International Conference on Computer Graphics and Interactive Techniques, pp. 401-408, 1999.

[110] A. O. Frank, A. Twombly, T. J. Barth*, et al.*, "Finite Element Methods for Real-time Haptic Feedback of Soft-Tissue Models in Virtual Reality Simulators," presented at Virtual Reality, 2001. Proceedings. IEEE, pp. 257-263, 2001.

[111] A. Petersik, B. Pflesser, U. Tiede*, et al.*, "Haptic Volume Interaction with Anatomic Models at Sub-Voxel Resolution," presented at Proceedings of the 10th Symp. On Haptic Interfaces for Virtual Environments & Teleoperator Systems (HAPTICS'02), pp. 66-72, 2002.

[112] C. Basdogan, S. De, J. Kim*, et al.*, "Haptics in Minimally Invasive Surgical Simulation and Training," *IEEE Computer Graphics and Applications*, vol. 24, pp. 56-64, 2004.

[113] S. L. Dawson and J. A. Kaufman, "The Imperative for Medical Simulation," *Proceedings of the IEEE*, vol. 86, pp. 479-483, 1998.

[114] D. Garroway and F. Bogsanyi, "6DOF Haptic Rendering using Geometric Algebra," *Haptic Virtual Environments and Their Applications, IEEE International Workshop (HAVE)*, pp. 103-107, 2002.

[115] M. Häggqvist and A. Lundqvist, "Mental Grasp: The development of a multi-sensory Mental Rotation Test platform." Masters, Umea University, 2003.

[116] N. W. John, N. Thacker, M. Pokric*, et al.*, "An Integrated Simulator for Surgery of the Petrous Bone," presented at Medicine Meets Virtual Reality 2001, pp. 218-224, 2001.

[117] G. J. Wiet, J. Bryan, E. Dodson*, et al.*, "Virtual Temporal Bone Dissection Simulation," presented at Medicine Meets Virtual Reality, pp. 378-384, 2000.

[118] J. Bryan, D. Stredney, D. G. Wiet, *et al.*, "Virtual Temporal Bone Dissection: A Case Study," presented at Conference on Visualization, pp. 497-500, 2001.

[119] A. Petersik, B. Pflesser, U. Tiede, *et al.*, "Haptic Rendering of Volumetric Anatomic Models at Sub-voxel Resolution," presented at Eurohaptics 2001, pp. 182-184, 2001.

[120] A. Petersik, B. Pflesser, U. Tiede, *et al.*, "Realistic Haptic Interaction in Volume Sculpting for Surgery Simulation," presented at Surgery Simulation and Soft Tissue Modeling, International Symposium (IS4TH 2003), pp. 194-202, 2003.

[121] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "Tracking the Movement of Surgical Tools in a Virtual Temporal Bone Dissection Simulator," presented at Surgery Simulation and Soft Tissue Modeling, pp. 102-109, 2003.

[122] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "Adaptive Techniques for Real-time Haptic and Visual Simulation of Bone Dissection," presented at Proceedings of the IEEE Virtual Reality 2003, pp. 1-8, 2003.

[123] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "A Mulitprocessor Decoupled System for the Simulation of Temporal Bone Surgery," *Computing and Visualization in Science*, vol. 5, pp. 35-43, 2002.

[124] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "Real-Time Haptic and Visual Simulation of Bone Dissection," *Presence*, vol. 12, pp. 110-122, 2003.

[125] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "A Haptic Model of a Bone-Cutting Burr," presented at Medicine Meets Virtual Reality, pp. 4-10, 2003.

[126] M. Hutchins, S. O'Leary, D. Stevenson, *et al.*, "A Networked Haptic Virtual Environment for Teaching Temporal Bone Surgery," presented at Medicine Meets Virtual Reality (MMVR) 13, pp. 204-207, 2005.

[127] U. Kühnapfel, H. Çakmak, B. Chantier, *et al.*, "HapticIO: Haptic Interface-Systems for Virtual-Reality Training in Minimally-Invasive Surgery," presented at International status conference Virtual One and Extended Reality, Leipzig, 2004.

[128] A. Pommert, K. H. Höhne, E. Burmester, *et al.*, "Computer-Based Anatomy: A Prerequisite for Computer-Assisted Radiology and Surgery," *Academic Radiology*, vol. 13, pp. 104-112, 2006.

[129] A. Petersik, B. Pflesser, U. Tiede, *et al.*, "Realistic Haptic Volume Interaction for Petrous Bone Surgery Simulation," presented at Computer Assisted Radiology and Surgery (CARS) Proceedings, Paris, France, 2002.

[130] M. Agus, A. Giachetti and E. Gobbetti, "Haptic Simulation of Bone Dissection," presented at Symposium on Methods and Applications in Advanced Computational Mechanics (SIMAI 2002), 2002.

[131] M. Agus, A. Giachetti, E. Gobbetti, *et al.*, "Mastoidectomy Simulation with Combined Visual and Haptic Feedback," *Studies in Health Technology and Informatics*, vol. 85, pp. 17-23, 2002.

[132] S. P. DiMaio and S. E. Salcudean, "Needle Insertion Modelling and Simulation," presented at IEEE International Conference on Robotics and Automation (ICRA), pp. 864-875, 2003.

[133] K.-U. Kyung, D.-S. Kwon, S.-M. Kwon, *et al.*, "Force Feedback for a Spine Biopsy Simulator with Volume Graphic Model," presented at IEEE International Conference on Intelligent Robots and Systems, pp. 1732-1737, 2001.

[134] D.-S. Kwon, K.-U. Kyung, S. M. Kwon, *et al.*, "Realistic Force Reflection in a Spine Biopsy Simulator," presented at IEEE International Conference on Robotics and Automation (ICRA), pp. 1358-1363, 2001.

[135] E. Gobbetti, M. Tuveri, G. Zanetti, *et al.*, "Catheter Insertion Simulation with co-registered Direct Volume Rendering and Haptic Feedback," presented at Medicine Meets Virtual Reality 2000 - Envisioning Healing: Interactive Technology and the Patient-Practioner Dialogue, pp. 96-98, 2000.

[136] L. Hiemenz, D. J. McDonald, D. Stredney, *et al.*, "A Physiologically Valid Simulator for Training Residents to Perform an Epidural Block," presented at Biomedical Engineering Conference, 1996, Proceedings of the 1996 Fifteenth Southern, pp. 170-172, 1996.

[137] X. Wang and A. Fenster, "A Virtual Reality Based 3D Real-Time Interactive Brachytherapy Simulation of Needle Insertion and Seed Implantation," presented at IEEE International Symposium on Biomedical Imaging: Macro to Nano, pp. 280-283, 2004.

[138] S. P. DiMaio and S. E. Salcudean, "Simulated Interactive Needle Insertion," presented at 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2002), pp. 344 - 351, 2002.

[139] W. Chou and T. Wang, "Human-Computer Interactive Simulation for the Training of Minimally Invasive Neurosurgery," presented at IEEE International Conference on Systems, Man and Cybernetics, pp. 1110-1115, 2003.

[140] S. P. DiMaio and S. E. Salcudean, "Interactive Simulation of Needle Insertion Models," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 1167-1179, 2005.

[141] P.-A. Heng and T.-T. Wong, "Intelligent Inferencing and Haptic Simulation for Chinese Acupuncture Learning and Training," *IEEE Transactions on Information Technology in Biomedicine*, pp. 1-1, 2005.

[142] F. P. Vidal, N. Chalmers, D. A. Gould, *et al.*, "Developing a Needle Guidance Virtual Environment with Patient-Specific Data and Force Feedback," *International Congress Series*, vol. 1281, pp. 418-423, 2005.

[143] A. Zorcolo, E. Gobbetti, P. Pili, *et al.*, "Catheter Insertion Simulation with Combined Visual and Haptic Feedback," presented at Proceedings First PHANToM Users Research Symposium (PURS'99), Heidelberg, Germany, 1999.

[144] A. Zorcolo, E. Gobbetti, G. Zanetti, *et al.*, "A Volumetric Virtual Environment for Catheter Insertion Simulation," presented at Virtual Environments 2000, Proceedings of the Eurographics Workshop, Amsterdam, The Netherlands, 2000.

[145] P. Gorman, T. Krummel, R. Webster, *et al.*, "A Prototype Haptic Lumbar Puncture Simulator," presented at Medicine Meets Virtual Reality (MMVR), pp. 106-109, 2000.

[146] G. Burdea, G. Patounakis, V. Popescu, *et al.*, "Virtual Reality-Based Training for the Diagnosis of Prostate Cancer," *IEEE Transactions on Biomedical Engineering*, vol. 46, pp. 1253-1260, 1999.

[147] G. Aloisio, L. Barone, M. Bergamasco, *et al.*, "Computer-Based Simulator for Catheter Insertion Training," presented at Medicine Meets Virtual Reality 12, pp. 4-6, 2004.

[148] M. Brahim and Y. Amirat, "Interactive Navigation Control with Haptic Rendering of Endovascular Treatment," presented at IEEE Conference on Robotics, Automation and Mechatronics, pp. 60-64, 2004.

[149] D. d'Aulignac, R. Balaniuk and C. Laugier, "A Haptic Interface for a Virtual Exam of the Human Thigh," presented at IEEE International Conference on Robotics and Automation (ICRA), pp. 2452-2457, 2000.

[150] P.-A. Heng, C.-Y. Cheng, T.-T. Wong, *et al.*, "A Virtual Reality based System for Training on Knee Arthroscopic Surgery," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 217-227, 2004.

[151] G. Zhang, S. Zhao and Y. Xu, "A Virtual Reality Based Arthroscopic Surgery Simulator," presented at IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, pp. 272 - 277, 2003.

[152] O. Korner and R. Manner, "Implementation of a Haptic Interface for a Virtual Reality Simulator for Flexible Endoscopy," presented at Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS), pp. 278-284, 2003.

[153] O. Korner, K. Rieger and R. Manner, "Haptic Display for all Degrees of Freedom of a Simulator for Flexible Endoscopy," presented at International Symposium on Medical Simulation, pp. 278-284, 2004.

[154] A. Neubauer, L. Mroz, S. Wolfsberger, *et al.*, "STEPS - An Application for Aimulation of Transsphenoidal Endonasal Pituitary Surgery," presented at IEEE conference on Visualization, pp. 513-520, 2004.

[155] A. Neubauer, S. Wolfsberger, M.-T. Forster, *et al.*, "Advanced Virtual Endoscopic Pituitary Surgery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, pp. 497 - 507, 2005.

[156] E. Papadopoulos, A. Tsamis and K. Vlachos, "A Real-Time Graphic Environment for a Urological Operation Training Simulator," *IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 2, pp. 1295 - 1300, 2004.

[157] G. Gopalakrishnan and V. Devarajan, "StapSim: A Virtual Reality-based Stapling Simulator for Laparoscopic Herniorrhaphy," presented at Medicine Meets Virtual Reality 12, Newport Beach, California, pp. 111-113, 2004.

[158] C. Basdogan, C.-H. Ho and M. A. Srinivasan, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct," *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp. 269-285, 2001.

[159] H. Zhang, S. Payandeh, J. Dill, *et al.*, "Acquiring Laparoscopic Manipulative Skills: A Virtual Tissue Dissection Training Module," presented at Medicine Meets Virtual Reality 12, pp. 419-421, 2004.

[160] C. Basdogan, C.-H. Ho and M. A. Srinivasan, "Simulation of Tissue Cutting and Bleeding for Laparoscopic Surgery Using Auxiliary Surfaces," presented at Medicine Meets Virtual Reality, pp. 38-44, 1999.

[161] C. Gunn, M. Hutchins, M. Adcock, *et al.*, "Surgical Training Using Haptics Over Long Internet Distances," presented at Medicine Meets Virtual Reality 12, pp. 121-123, 2004.

[162] M. P. Ottensmeyer, E. Ben-Ur and J. K. Sailsbury, "Input and Output for Surgical Simulation: Devices to Measure Tissue Properties in vivo and a Haptic Interface for Laparoscopy Simulators," presented at Medicine Meets Virtual Reality, pp. 236-242, 2000.

[163] H. Maaß, H. Çakmak, U. Kühnapfel, *et al.*, "Providing More Possibilities for Haptic Devices in Surgery Simulation," *International Congress Series*, vol. 1281, pp. 725-729, 2005.

[164] M. Bro-Nielsen, D. Helfrick, B. Glass, *et al.*, "VR Simulation of Abdominal Trauma Surgery," presented at Medicine Meets Virtual Reality (MMVR), pp. 117-123, 1998.

[165] A. Chanda and T. Kesavadas, "Real-Time Volume Haptic Rendering of Non-linear Viscoelastic Behavior of Soft Tissue through Dynamic Atomic Unit Approach," presented at Medicine Meets Virtual Reality 12, Newport Beach, California, pp. 16-17, 2004.

[166] Y. Kuroda, M. Nakao, T. Kuroda, *et al.*, "Interaction Model Between Elastic Objects for Haptic Feedback Considering Collisions of Soft Tissue," *Computer Methods and Programs in Biomedicine*, vol. 80, pp. 216-224, 2005.

[167] Y.-J. Lim and S. De, "Nonlinear Tissue Response Modeling for Physically Realistic Virtual Surgery using PAFF," presented at First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC 2005), pp. 479-480, 2005.

[168] V. Daniulaitis, M. O. Alhalabi, H. Kawasaki, *et al.*, "Medical Palpation of Deformable Tissue using Physics-Based Model for Haptic Interface Robot (HIRO)," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), pp. 3907-3911, 2004.

[169] M. O. Alhalabi, V. Daniulaitis, H. Kawasaki, *et al.*, "Medical Training Simulation for Palpation of Subsurface Tumor using HIRO," presented at First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC 2005), pp. 623-624, 2005.

[170] A. J. Lindblad, G. M. Turkiyyah, G. Sankaranarayanan, *et al.*, "Two-Handed Next Generation Suturing Simulator," presented at Medicine Meets Virtual Reality 12, pp. 215-220, 2004.

[171] I. F. Costa and R. Balaniuk, "LEM- An Approach for Real Time Physically Based Soft Tissue Simulation," presented at Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Seoul, Korea, pp. 2337-2343, 2001.

[172] S. Cotin, H. Delingette and N. Ayache, "Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 62-73, 1999.

[173] N. Suzuki, A. Hattori, S. Suzuki, *et al.*, "Virtual Surgery System with Haptic Sensation for Both Hands," presented at Proceedings of the 22nd Annual EMBS International Conference, Chicago IL, pp. 2419-2422, 2000.

[174] R. J. Lapeer, P. Gasson, J.-L. Florens, *et al.*, "Introducing a Novel Haptic Interface for the Planning and Simulation of Open Surgery," presented at Medicine Meets Virtual Reality 12, Newport Beach, California, pp. 197-199, 2004.

[175] D. J. Weiss and A. M. Okamura, "Haptic Rendering of Tissue Cutting with Scissors," presented at Medicine Meets Virtual Reality 12, Newport Beach, California, pp. 407-409, 2004.

[176] A. Benali, P. Richard and P. Bidaud, "Design, Control and Evaluation of a Six DOF Force Feedback Interface for Virtual Reality Applications," presented at Proceedings of the 1999 IEEE International Workshop on Robot and Human Interaction, Pisa, Italy, pp. 338-343, 1999.

[177] Y.-J. Lim and S. De, "Realistic Simulation of Surgical Cutting of Soft Tissues in Real Tim with Force Feedback," presented at Medicine Meets Virtual Reality 12, pp. 210-214, 2004.

[178] A. Faraci, F. Bello and A. Darzi, "Soft Tissue Deformation using a Hierarchical Finite Element Model," presented at Medicine Meets Virtual Reality 12, pp. 92-98, 2004.

[179] R. Webster, J. Sessani, R. Shenk, *et al.*, "A Haptic Surgical Simulator for the Continuous Curvilinear Capsulorhexis Procedure During Cataract Surgery," presented at Medicine Meets Virtual Reality 12, Newport Beach, California, pp. 404-406, 2004.

[180] K.-S. Choi, H. Sun and P.-A. Heng, "Interactive Deformation of Soft Tissues with Haptic Feedback for Medical Learning," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, pp. 358-363, 2003.

[181] D. G. Wiet, "Validation/Dissemination of Virtual Temporal Bone Dissection," Biomedical Applications Research Group, 2006, sited 01_01_06, http://www.osc.edu/research/Biomed/projects/vtbone/index.shtml.

[182] H. Delingette, S. Cotin and N. Ayache, "A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation," presented at Computer Animation, 1999 Proc., pp. 1-12, 1999.

[183] V. Vuskovic, M. Kauer, G. Szekely, *et al.*, "Realistic Force Feedback for Virtual Reality Based Diagnostic Surgery Simulators," presented at IEEE International Conference on Robotics and Automation (ICRA), pp. 1592-1598, 2000.

[184] S. Cotin, H. Delingette and N. Ayache, "Real Time Volumetric Deformable Models for Surgery Simulation," presented at 4th International Conference Visualization in Biomedical Computing (VBC), pp. 535-540, 1996.

[185] S. Schendel, K. Montgomery, A. Sorokin, *et al.*, "A Surgical Simulator for Planning and Performing Repair of Cleft Lips," *Cranio-Maxillo-Facial Surgery*, vol. 33, pp. 223-228, 2005.

[186] C. Mendoza and C. Laugier, "Simulating Cutting in Surgery Applications using Haptics and Finite Element Models," presented at IEEE Virtual Reality Conference (VR '03), pp. 295, 2003.

[187] D. J. Weiss and A. M. Okamura, "Haptic Rendering of Tissue Cutting with Scissors," presented at Medicine Meets Virtual Reality (MMVR), pp. 407-409, 2004.

[188] H. Zhang, S. Payandeh and J. Dill, "On Cutting and Dissection of Virtual Deformable Objects," presented at IEEE International Conference on Robotics and Automation (ICRA '04), pp. 3908-3913, 2004.

[189] W. J. Ahn and D. Y. Lee, "Interactive Haptic Modeling of Colon and Colonoscope," presented at Medicine Meets Virtual Reality 12, pp. 1-3, 2004.

[190] U. Kühnapfel, H. K. Çakmak and H. Maaß, "Endoscopic Surgery Training using Virtual Reality and Deformable Tissue Simulation," *Computers & Graphics*, vol. 24, pp. 671-682, 2000.

[191] H. K. Çakmak and U. Kühnapfel, "Animation and Simulation Techniques for VR-Training Systems in Endoscopic Surgery," presented at Eurographics Workshop on Animation and Simulation '2000 (EGCAS '2000), pp. 173-185, 2000.

[192] U. Kühnapfel, H. K. Çakmak and H. Maass, "3D Modeling for Endoscopic Surgery," presented at Proceedings of the IEEE Symposium on Simulation, pp. 22-32, 1999.

[193] N. Nurdin, E. Weilandt, M. Textor, *et al.*, "Reduced Frictional Resistance of Poyurethane Catheter by Means of a Surface Coating Procedure," *Journal of Applied Polymer Science*, vol. 61, pp. 1939-1948, 1996.

[194] A. Galliano, S. Bistac and J. Schultz, "Adhesion and Friction of PDMS Networks: Molecular Weight Effects," *Journal of Colloid Interface Science*, vol. 265, pp. 372-379, 2003.

[195] S.-I. Hatsushika, R. K. Shepherd, Y. C. Tong, *et al.*, "Dimensions of the Scala Tympani in the Human and Cat with Reference to Cochlear Implants," *Annals of Otology, Rhinology & Laryngology*, vol. 99, pp. 871-876, 1990.

[196] J. Wysocki, "Dimensions of the Human Vestibular and Tympanic Scalae," *Hearing Research*, vol. 135, pp. 39-46, 1999.

[197] F. W. Zonneveld, "Computed Tomography of the Temporal Bone and Orbit Technique of Direct Multiplanar, High-resolution CT and Correlative Cryosectional Anatomy." Munich, Urban & Schwarzenberg, 1987.

[198] M. H. Bomans, Karl-Heinz; Tiede, Ulf; Riemer, Martin, "3-D Segmentation of MR Images of the Head for 3-D Display," *IEEE Transactions on Medical Imaging*, vol. 9, pp. 177-183, 1990.

[199] Med-El, 2004, sited 24_05_04, http://www.medel.com/ENG/INT/30_Advanced_topics/60_Human_Cochlear_Anatomy/000_Human_Cochlear_Anatomy.asp.

[200] H. M. Ladak, "Finite-Element Modelling of Middle-Ear Prostheses in Cats." Masters Thesis, Montreal, Canada, McGill University, 1993.

[201] R. Lakes, H. S. Yoon and J. L. Katz, "Ultrasonic Wave Propagation and Attenuation in Wet Bone," *Journal of Biomedical Engineering*, vol. 8, pp. 143-148, 1986.

[202] J. Raethjen, F. Pawlas, M. Lindemann, *et al.*, "Determinants of Physiologic Tremor in a Large Normal Population," *Clinical Neurophysiology*, vol. 111, pp. 1825-1837, 2000.

[203] TableCurve2D, " Systat Software, Inc.," sited 20_02_06, http://www.systat.com/downloads/?sec=d002.

[204] K. V. Nesbitt, "Structured Guidelines to Support the Design of Haptic Displays," presented at Guidance on Tactile and Haptic Interactions (GOTHI'05), Saskatoon, Saskatchewan, Canada, 2005.

Appendix A

Cochlear Surface Extraction: VTK/Python

#filename: Visualise.py

```
#image processing: avw to vtk format, output slices to interface
from vtkpython import *
from vtkpythontk import *
from Tkinter import *
from Viewers import *
from ReadAVWModel import *
ctimage =
read_avw_model("Volume.avw",(0,511,0,511,0,294),(0,0,0),(0.00021875,0.00021875,0.00021875))

#crop image to decrease spatial resolution
imclip1 = vtkImageClip()
imclip1.SetInput(ctimage)
imclip1.SetOutputWholeExtent(100,350,130,335,0,255)
info1 = vtkImageChangeInformation()
info1.SetInput(imclip1.GetOutput())
info1.CenterImageOn()
info1.SetOutputExtentStart(0,0,0)

#scale image to one byte/pixel
ss1 = vtkImageShiftScale()
ss1.SetInput(info1.GetOutput())
ss1.SetShift(0.0)
ss1.SetScale(255.0/2000)
ss1.SetOutputScalarTypeToUnsignedChar()
ss1.ClampOverflowOn()
permute1 = vtkImagePermute()
permute1.SetInput(ss1.GetOutput())
permute1.SetFilteredAxes(0,2,1)
flip1 = vtkImageFlip()
flip1.SetInput(permute1.GetOutput())
flip1.SetFilteredAxis(0)
flip2 = vtkImageFlip()
flip2.SetInput(flip1.GetOutput())
flip2.SetFilteredAxis(2)
flip2.Update()
writer1 = vtkStructuredPointsWriter()
writer1.SetInput(flip2.GetOutput())
writer1.SetFileName("EarR1.vtk")
writer1.SetFileTypeToBinary()
writer1.Write()
tk = Tk()
win1 = make_image_viewer(tk,flip2.GetOutput())
quit_b = Button(tk)
quit_b.pack(side="bottom",fill="x",expand="no")
quit_b["text"] = "Quit"
quit_b["command"] = tk.quit
tk.mainloop()
```

#filename: Binary.py

```
#segmentation, cochlea
from vtkpython import *
from vtkpythontk import *
```

```
from Tkinter import *

from Viewers import *
from SegmentationViewer import *
reader1 = vtkStructuredPointsReader()
reader1.SetFileName("EarR1.vtk")
reader1.Update()
seg_image = vtkImageData()
seg_image.SetScalarTypeToUnsignedChar()
seg_image.CopyStructure(reader1.GetOutput())
seg_image.AllocateScalars()
writer1 = vtkStructuredPointsWriter()
writer1.SetInput(seg_image) #data object, not process object in pipeline
writer1.SetFileName("SegEarR1.vtk")
writer1.SetFileTypeToBinary()
writer1.Write()
tk = Tk()
win1 = make_image_viewer(tk,seg_image)
quit_b = Button(tk)
quit_b.pack(side="bottom",fill="x",expand="no")
quit_b["text"] = "Quit"
quit_b["command"] = tk.quit
tk.mainloop()
```

#filename: Segment.py

```
#segmentation, cochlea
from vtkpython import *
from vtkpythontk import *
from Tkinter import *
from Viewers import *
from SegmentationViewer import *
reader1 = vtkStructuredPointsReader()
reader1.SetFileName("EarR1.vtk")
reader1.Update()
reader2 = vtkStructuredPointsReader()
reader2.SetFileName("SegEarR1.vtk")
reader2.Update()
ct_image = vtkImageData()
ct_image.DeepCopy(reader1.GetOutput())
seg_image = vtkImageData()
seg_image.DeepCopy(reader2.GetOutput())
tk = Tk()
viewer1 = SegmentationViewer(tk,ct_image,seg_image,
  ["cochlea","boundary"],"SegEarR1.vtk")
quit_b = Button(tk)
quit_b.pack(side="bottom",fill="x",expand="no")
quit_b["text"] = "Quit"
quit_b["command"] = tk.quit
tk.mainloop()
```

#filename: Render.py

```
#import vtk file (segmentation file, previously developed, with Segment.py) and
#export as VRML file: surface description of cochlea
from vtkpython import *
from vtkpythontk import *
from Tkinter import *
from Viewers import *
from SegmentationViewer import *
```

```
#read in vtk file
reader1 = vtkStructuredPointsReader()
reader1.SetFileName("SegEarR1.vtk") #need cochlea object!
reader1.Update()
reader2 = vtkStructuredPointsReader()
reader2.SetFileName("EarR1.vtk")
reader2.Update()
ct_image = vtkImageData()
ct_image.DeepCopy(reader2.GetOutput())

#from vtk file obtain cochlea object to make vrml surface
thresh1 = vtkImageThreshold()
thresh1.SetInput(reader1.GetOutput())
thresh1.SetOutputScalarTypeToUnsignedChar()
thresh1.ThresholdBetween(1,1)  #looking for cochlea, labelled 1
thresh1.ReplaceInOn()
thresh1.SetInValue(255)
thresh1.ReplaceOutOn()
thresh1.SetOutValue(0)

#create vrml surface
cast1 = vtkImageCast()
cast1.SetInput(thresh1.GetOutput())
cast1.SetOutputScalarTypeToFloat()
smooth1 = vtkImageGaussianSmooth()
smooth1.SetInput(cast1.GetOutput())
smooth1.SetStandardDeviations(1,1,1)
smooth1.SetRadiusFactors(3,3,3)
smooth1.SetDimensionality(3)
con1 = vtkContourFilter()  #convert volume data to surface data
con1.SetInput(smooth1.GetOutput())
con1.SetValue(0,128)
con1.ComputeNormalsOn()
out_polys = con1.GetOutput()   #changed from norm1, creates object
out_polys.Update()
tk = Tk()

#view data: mixed gives CT plus polygonal representation
win1 = make_mixed_viewer(tk,ct_image,(out_polys,(0,0,1))) #to change color, out_polys is simply grey

#write to VRML file
mapper1 = vtkPolyDataMapper()
mapper1.SetInput(out_polys)
mapper1.ScalarVisibilityOff()
actor1 = vtkActor()
actor1.SetMapper(mapper1)
ren1 = vtkRenderer()
ren1.AddActor(actor1)
win2 = vtkRenderWindow() #viewing window
win2.AddRenderer(ren1)
vrml1 = vtkVRMLExporter()
vrml1.SetInput(win2)
vrml1.SetFileName("SetCochleaVRML.wrl")
vrml1.Write()
print "Wrote file: SetCochleaVRML.wrl"
quit_b = Button(tk)
quit_b.pack(side="bottom",fill="x",expand="no")
quit_b["text"] = "Quit"
quit_b["command"] = tk.quit
tk.mainloop()
```

Appendix B

Cochlear VRML Surface Visualisation in the Reachin API

```
Display {
children [
DEF MYTRANS Transform {
rotation 0 0 0 0   #no rotation at start. want to access this field to rotate structure interactively
scale 4 4 4
children [
Shape {
appearance Appearance {
material Material {
ambientIntensity 0
diffuseColor 1 1 1
specularColor 0 0 0
shininess 0.0078125
transparency 0
}
surface SimpleSurface {}
}
geometry IndexedFaceSet {
solid FALSE
coord DEF VTKcoordinates Coordinate {
point [
-0.01225 0.0079844 0.00250836,
-0.01225 0.00797328 0.0025156,
-0.0122834 0.0079844 0.0025156,
…
-0.0168438 0.00951565 0.0148414,
-0.0166251 0.00951565 0.0147923,
]}
coordIndex  [
0, 1, 2, -1,
0, 3, 4, -1,
…
14723, 14701, 14722, -1,
14689, 14701, 14723, -1,
]} #end Geometry
} #end Shape
]} #end Transform
]} #end Display

#python script code
DEF ROTATECOCHLEA PythonScript {
  url "RotateBoneKeyR1.py"
  references [ USE MYTRANS  ]
}
```

Appendix C

Cochlea Rotation and Translation

#filename: RotateBoneKeyR1.py

#function to rotate structure in Reachin using Keyboard commands

```
import Keyboard
MyTr1 = references[0]     #MYTRANS overall transform from wrl to this python file

class KeyboardCatcher(Dependent(SFString)):
    def evaluate(self,inputs):
        #ROTATE UP
        if inputs.get() == 'i':     #note capitals...shift i reqd.
            angle_inc = -0.1745 #decrease angle 10 degrees in rads
            Res1 = Rotation(Vec3f(1,0,0),angle_inc)*MyTr1.rotation.get()
            MyTr1.rotation.set(Res1)  #x,y,z,angle_inc; rot. about x axis
        #ROTATE DOWN
        elif inputs.get() == 'k':
            angle_inc = +0.1745 #increase angle 10 degrees in rads
            Res1 = Rotation(Vec3f(1,0,0),angle_inc)*MyTr1.rotation.get()
            MyTr1.rotation.set(Res1)  #x,y,z,angle_inc; rot. about x axis
        #ROTATE LEFT
        elif inputs.get() == 'j':
            angle_inc = -0.1745 #decrease angle 10 degrees in rads
            Res1 = Rotation(Vec3f(0,1,0),angle_inc)*MyTr1.rotation.get()
            MyTr1.rotation.set(Res1)  #x,y,z,angle_inc; rot. about y axis
        #ROTATE RIGHT
        elif inputs.get() == 'l':
            angle_inc = 0.1745 #increase angle 10 degrees in rads
            Res1 = Rotation(Vec3f(0,1,0),angle_inc)*MyTr1.rotation.get()
            MyTr1.rotation.set(Res1)  #x,y,z,angle_inc; rot. about y axis
        #TRANSLATE LEFT
        elif inputs.get() == 'u':
            #set translation to left
            MyTr1.translation.set(MyTr1.translation.get() + Vec3f(-0.08, 0, 0)) #move left along x-
axis
        #TRANSLATE RIGHT
        elif inputs.get() == 'o':
            #set translation to right
            MyTr1.translation.set(MyTr1.translation.get() + Vec3f(0.08, 0, 0)) #move right along
x-axis

#instantiation of class
catcher = KeyboardCatcher()
Keyboard.characters.route(catcher)
```

Appendix D

Custom Object Nodes

#filename: CochlearV.h

```
#ifndef COCHLEARV_H
#define COCHLEARV_H

#include <ForceTorqueGroup.h>
#include <Group.h>
#include <Vrml.h>
#include "STympH.h"
#include "STympV.h"
#include "CArr.h"
#include "Stylet.h"
#include "BMembV.h"

using namespace Reachin;

class CochlearV : public Group {
protected:

  //make sure nodes are correct type, when entered into VRML file
  struct SFST : public TypedMFNode< ForceTorqueGroup > { //STymp
  };
  struct SFCA : public TypedMFNode< CArr > {
  };
  struct SFS : public TypedMFNode< Stylet > { //Stylet
  };
  struct SFBM : public TypedMFNode< BMembV > {
  };
  struct CochlearComposer : public EvaldFField< CochlearComposer, MFNode,
                                                SFCA, SFS, SFBM, SFST > {
          void evaluate( SFCA *cont, SFS *styl, SFBM *memb, SFST *tymp );
  };

public:
  CochlearV();

  static const Interface interface;

  auto_ptr< SFST > tymp;
  auto_ptr< SFCA > cont;
  auto_ptr< SFS  > styl;
  auto_ptr< SFBM > memb;

protected:

  auto_ptr< CochlearComposer > cochlear_composer;
  DEFMap defmapB;

};

#endif
```

#filename: CochlearV.cpp

```
#include <Transform.h>
#include <GraspableTransform.h>
#include "CochlearV.h"

const Interface CochlearV::interface(
   "CochlearV",
   typeid(CochlearV), Create<CochlearV>::create,
   exposedField( "tymp", &CochlearV::tymp ) +
   exposedField( "cont", &CochlearV::cont ) +
   exposedField( "styl", &CochlearV::styl ) +
   exposedField( "memb", &CochlearV::memb )
   );

CochlearV::CochlearV() :
  cont          ( new SFCA),
  styl          ( new SFS ),
  memb          ( new SFBM ),
  tymp          ( new SFST ),
  cochlear_composer    ( new CochlearComposer ) {

  cont->route( cochlear_composer );
  styl->route( cochlear_composer );
  memb->route( cochlear_composer );
  tymp->route( cochlear_composer );

  cochlear_composer->route( children );

}


void CochlearV::CochlearComposer::evaluate( SFCA *cont, SFS *styl, SFBM *memb,
                                                                    SFST *tymp ) {
  //exit all existing values
  Util::for_each( value.begin(), value.end(), this, &MFNode::exitNode );

  //MFNode to contain nothing
  value.resize(0);

  for( SFST::const_iterator i = tymp->begin();
               i != tymp->end();
               i++ ) {
            value.push_back( (*i));
  };

  for( SFCA::const_iterator j = cont->begin();
               j != cont->end();
               j++ ) {
            value.push_back( (*j));
  };

  for( SFS::const_iterator k = styl->begin();
               k != styl->end();
               k++ ) {
            value.push_back( (*k));
  };

  for( SFBM::const_iterator l = memb->begin();
               l != memb->end();
               l++ ) {
            value.push_back( (*l));
```

```
  };

  //Enter all new values
  Util::for_each( value.begin(), value.end(), this, &MFNode::enterNode );
}
```

#filename: STympV.h

```
#ifndef STYMPV_H
#define STYMPV_H

#include <Transform.h>

using namespace Reachin;

class STympV: public Transform {
public:

  static const Interface interface;

  STympV(); //default constructor for scala tympani

};

#endif
```

#filename: STympV.cpp

```
#include <Vrml.h>
#include <Transform.h>
#include "STympV.h"

const Interface STympV::interface (
  "STympV",
  typeid(STympV), Create<STympV>::create,
  exposedField( "translation", &STympV::translation  ) +
  exposedField( "scale", &STympV::scale     )
);

STympV::STympV() {

  DEFMap def_mapST;
  createVrmlFromURL("reduce_ST_r3_crop.wrl", &def_mapST);   //ScalaTympOnly.wrl
  Transform *tfst;
  def_mapST.find("MYTRANS", tfst);

  children->add(tfst);

  //set translation and scale
  translation->set(Vec3f(0, 0, 0));
  scale->set(Vec3f(0.001, 0.001, 0.001)); //Vec3f(0.02, 0.02, 0.02)

}
```

#filename: STympH.h

```
#ifndef STYMPH_H
#define STYMPH_H

#include <Transform.h>

using namespace Reachin;

class STympH: public Transform {
public:

  static const Interface interface;

  STympH(); //default constructor for scala tympani

};

#endif
```

#filename: STympH.cpp

```
#include <Vrml.h>
#include <Transform.h>
#include "STympH.h"

const Interface STympH::interface (
  "STympH",
  typeid(STympH), Create<STympH>::create,
  exposedField( "translation", &STympH::translation  ) +
  exposedField( "scale", &STympH::scale     )
);

STympH::STympH() {

  DEFMap def_mapST;
  createVrmlFromURL("vwaif2_95.wrl", &def_mapST);
  Transform *tfst;
  def_mapST.find("MYTRANS", tfst);

  children->add(tfst);

  //set translation and scale
  translation->set(Vec3f(0, 0, 0)); //translation and scale for Scala Tymp and BM, was -0.2 to move back
in z-dir, Vec3f(0.04, 0, 0)
  scale->set(Vec3f(0.001, 0.001, 0.001)); //Vec3f(0.02, 0.02, 0.02)

}
```

#filename: Stylet.h

```
#ifndef STYLET_H
#define STYLET_H

#include <Transform.h>
```

```
using namespace Reachin;

class Stylet: public Transform {
public:

  static const Interface interface;

  Stylet(); //default constructor for stylet

  auto_ptr< SFVec3f > inner_trans;

};

#endif
```

#filename: Stylet.cpp

```
#include <Vrml.h>
#include <Transform.h>
#include <Field.h>
#include <SField.h>
#include <MField.h>
#include <ForceTorqueGroup.h>
#include "Stylet.h"

const Interface Stylet::interface (
  "Stylet",
  typeid(Stylet), Create<Stylet>::create,
  exposedField( "translation", &Stylet::translation      ) +
  exposedField( "rotation", &Stylet::rotation            ) +
  exposedField( "scale", &Stylet::scale                          ) +
  exposedField( "inner_trans", &Stylet::inner_trans    )
//  exposedField( "pos", &Stylet::global_grasp_position          ) + //exposed in VRML and routed to
Display
//  exposedField( "orn", &Stylet::global_grasp_orientation) +
//  exposedField( "grp", &Stylet::grasp                          )
);

Stylet::Stylet():
  inner_trans        ( new SFVec3f ) {

  DEFMap def_mapS;
  createVrmlFromURL("StyletCROPFinal70.wrl", &def_mapS); //StyletSurfaceOnly.wrl
stylet_remesh_final.wrl
  Transform *tfs;
  def_mapS.find("GTST", tfs);

  inner_trans->route(def_mapS.find("INNER_T.translation")); //route prev
// inner_trans->touch();

  children->add(tfs);

}
```

#filename: BMembV.h

```
#ifndef BMEMBV_H
#define BMEMBV_H

#include <Transform.h>

using namespace Reachin;

class BMembV: public Transform {
public:

  static const Interface interface;

  BMembV(); //default constructor for basilar membrane

};

#endif
```

#filename: BMembV.cpp

```
#include <Vrml.h>
#include <Transform.h>
#include "BMembV.h"

const Interface BMembV::interface (
  "BMembV",
  typeid(BMembV), Create<BMembV>::create,
  exposedField( "translation", &BMembV::translation) +
  exposedField( "scale", &BMembV::scale    )
);

BMembV::BMembV() {

  DEFMap def_mapBM;
  createVrmlFromURL("reduce_BM_r2_crop.wrl", &def_mapBM); //after VWaif, appended _crop
  Transform *tfbm; //create new instance of Transform nodes
  def_mapBM.find("MYTRANS", tfbm);

  children->add(tfbm);

  //set translation and scale
  translation->set(Vec3f(0, 0, 0));
  scale->set(Vec3f(0.001, 0.001, 0.001));

}
```

#filename: CArr.h

```
#ifndef CARR_H
#define CARR_H

#include <Transform.h>

using namespace Reachin;

class CArr: public Transform {
```

```
public:

  static const Interface interface;

  CArr(); //default constructor for contour array

  //create public fields for access in VRML
//  auto_ptr <SFVec3f> pos;
//  auto_ptr <SFRotation> orn;
//  auto_ptr <SFBool> grp;

};

#endif
```

#filename: CArr.cpp

```
#include <Vrml.h>
#include <Transform.h>
#include "CArr.h"

const Interface CArr::interface (
  "CArr",
  typeid(CArr), Create<CArr>::create,
  exposedField( "translation", &CArr::translation       ) +
  exposedField( "rotation", &CArr::rotation             ) +
  exposedField( "scale", &CArr::scale                           )
//  exposedField( "pos", &CArr::global_grasp_position         ) + //exposed in VRML and routed to
Display
//  exposedField( "orn", &CArr::global_grasp_orientation      ) +
//  exposedField( "grp", &CArr::grasp                         )
);

CArr::CArr() {

  DEFMap def_mapCA;

  createVrmlFromURL("ContElecArrFin85.wrl", &def_mapCA); //ContourArrayOnly.wrl;
Contour_remesh_final.wrl, ContElecArrFin70.wrl
  Transform *tfca;
  def_mapCA.find("GT", tfca);

  children->add(tfca);

  //set translation and scale fields
//  translation->set(Vec3f(0.04, 0, 0)); //translation and scale for Scala Tymp and BM
//  scale->set(Vec3f(0.01, 0.01, 0.01));

}
```

Appendix E

Insertion Scene

#filename: run_simulation10.wrl

```
DEF VIEWPT Viewpoint{
fieldOfView 0.03 #smaller makes the scene larger
}
#Background {
#color 0.4 0.4 0.4
#}
Import {
url [
"urn:inet:reachin.se:/display/tools/toolsCI/fout_final85.wrl", #fOUT_comb.wrl; forceps_out_final.wrl
"urn:inet:reachin.se:/display/tools/toolsCI/fin_final85.wrl", #fIN2_final.wrl; forceps_in_final.wrl
"urn:inet:reachin.se:/bin/UserInterface"
]}
DEF DIS Display{
tip Shape{}
stylus Transform {
children [
DEF SWSTYL Switch {
whichChoice 0
choice [
USE FOUT
USE FIN
]}]}
children [
DEF CON CollidingController {
mass 0.0178943 #measured: 0.00031kg elec only; add on tweezer weight: 17.587g; total: 0.0178943
proxyRadius 0.00025 #had 0.0005 then 0.00025 #proxyRadius 0.000125  #default: 0.0025, too large
springStiffness 350 #150 LATEST #default is 200 N/m; try from 15 N/m to 500 N/m
springDamping 1.0 #default is 1.0 Ns/m
rotSpringDamping 0.00003 #leave as default 0.0003
rotSpringStiffness 0.2 #leave as default
inertia BoxInertia {
size 0.03 0.02 0.02
mass 0.0178943 #measured: 0.00031kg elec only; add on tweezer weight: 17.587g; total: 0.0178943
}
children [
Shape {
appearance Appearance{
material DEF _v2%0 Material {
diffuseColor 1 1 1
emissiveColor 0.9 0.9 0.9
transparency 0.6
} }
geometry DEF PTSET PointSet {
#color 1 0 0
coord DEF PX_COORDS Coordinate {
point[]
}
pointSize 1.5 #25 #1.5, lower lim
}}]
points[]
collisionChild DEF CG Group {
children [
```

```
CochlearV {
tymp[
DEF FTSTYMP ForceTorqueGroup {
children [
DEF STAYST STympH {
scale 0.001 0.001 0.001 #0.02 0.02 0.02
}]}]}]}}
DEF BTN0 Transform {
translation -0.15 -0.075 0
children [
Frame {
children [
DEF BUTTON_A Button {
desiredDimensions 0.05 0.01 0.01
text "Tip Position"
}
DEF BUTTON_B Button {
desiredDimensions 0.05 0.01 0.01
}]}]}
DEF BTN1 Transform {
translation -0.15 -0.1 0
children [
Frame {
children [
DEF BUTTON_C Button {
desiredDimensions 0.05 0.01 0.01
text "Output Force"
}
DEF BUTTON_D Button {
desiredDimensions 0.05 0.01 0.01
}]}]}
DEF BTN2 Transform {
translation -0.15 -0.125 0
children [
Frame {
children [
DEF BUTTON_E Button {
desiredDimensions 0.05 0.01 0.01
text "Output Torque"
}
DEF BUTTON_F Button {
desiredDimensions 0.05 0.01 0.01
}]}]}
CochlearV {
#styl [
#DEF FTGSTS ForceTorqueGroup {
#children [
#DEF MOVES Stylet {
#translation -0.07 0 0
#scale 0.001 0.001 0.001
#}]
#}]
memb [
DEF STAYBM BMembV {
scale 0.001 0.001 0.001
}]
tymp[
DEF FTSTYMPV ForceTorqueGroup {
children [
DEF STAYSTV STympV {
```

```
scale 0.001 0.001 0.001 #0.02 0.02 0.02
}]}]
cont [
DEF MOVECA CArr {
translation -0.07 0 0
scale 0.001 0.001 0.001 #0.02 0.02 0.02 #
}]
}]}
DEF ZOOM PythonScript {
url "zoom7e.py"     #was zoom7.py
references [ USE VIEWPT, USE BTN0, USE BTN1, USE BTN2, USE PTSET ]
}
DEF MOVECARRIER PythonScript {
url "MoveCarrier_noCA11.py"
references [ USE MOVECA, USE DIS, USE SWSTYL, USE CON, USE PX_COORDS ]
}
#DEF MOVECARRIER PythonScript {
#url "MoveCarrier_noCA11.py"
#references [ USE DIS, USE SWSTYL, USE CON, USE PX_COORDS ] #USE MOVES
#}
#DEF SELECTTOOL PythonScript {
#url "keysR8.py" #without scale changes, instead of just keys.py
#references [ USE DIS, USE SWSTYL, USE STAYST, USE MOVECA, USE MOVES, USE STAYBM,
#USE FOUT, USE FIN, USE CON,USE PX_COORDS ]
#}
#DEF PS PythonScript {
#url "ForceT.py"
#references [ USE MOVES ]
#}
DEF PSA PythonScript {
url "ForceDataCapture29f.py" #14.py; 8.py originally, with initial insertion points #29a.py ;29d.py
references [ USE FTSTYMP, USE DIS, USE CON, USE PX_COORDS ]
}
ROUTE DIS.devicePosition TO PSA.pos_string #changed from trackerPosition
ROUTE PSA.pos_string TO BUTTON_B.text
ROUTE FTSTYMP.force TO PSA.force_string
ROUTE PSA.force_string TO BUTTON_D.text
ROUTE FTSTYMP.torque TO PSA.torque_string
ROUTE PSA.torque_string TO BUTTON_F.text
ROUTE CON.pointProxies TO PX_COORDS.point
```

#filename: zoom7e.py

```
#zoom.py will change magnification of scene, and
#change position of force/location o/p buttons, to always appear on screen with limits
#zoom values by incremental changes: different values for different levels of zoom

import Keyboard

vpoint = references[0]
butn0 = references[1]
butn1 = references[2]
butn2 = references[3]
ptgeom = references[4]

class KeyboardCatcher(Dependent(SFString)):
    def evaluate(self,inputs):
        #scene magnification: note that as fOV increases in value, the object size decreases and vice
versa
```

```
            if inputs.get() == '+':
                    #at start, Viewpoint FOV = 0.03
                    if vpoint.fieldOfView.get()==0.03:
                            #1st zoom in, set FOV, button translation and scale
                            vpoint.fieldOfView.set(vpoint.fieldOfView.get() - 0.5 *
vpoint.fieldOfView.get())
                            butn0.translation.set(Vec3f(butn0.translation.get().x + 0.075,
                            butn0.translation.get().y + 0.0375, butn0.translation.get().z )) #set btn pos
                            butn1.translation.set(Vec3f(butn1.translation.get().x + 0.075,
                            butn1.translation.get().y + 0.05, butn1.translation.get().z )) #set btn pos
                            butn2.translation.set(Vec3f(butn2.translation.get().x + 0.075,
                            butn2.translation.get().y + 0.0625, butn2.translation.get().z ))
                            #set button scale
                            butn0.scale.set(Vec3f(butn0.scale.get().x - 0.5005, butn0.scale.get().y - 0.5005,
                            butn0.scale.get().z - 0.5005)) #set btn scale
                            butn1.scale.set(Vec3f(butn1.scale.get().x - 0.5005, butn1.scale.get().y - 0.5005,
                            butn1.scale.get().z - 0.5005)) #set btn scale
                            butn2.scale.set(Vec3f(butn2.scale.get().x - 0.5005, butn2.scale.get().y - 0.5005,
                            butn2.scale.get().z - 0.5005))
                            ptgeom.pointSize.set(3.0)
                    elif vpoint.fieldOfView.get()==0.015:
                            #2nd zoom in, set FOV, button translation and scale
                            vpoint.fieldOfView.set(vpoint.fieldOfView.get() - 0.5 *
vpoint.fieldOfView.get())
                            butn0.translation.set(Vec3f(butn0.translation.get().x + 0.0375,
                            butn0.translation.get().y + 0.01875, butn0.translation.get().z )) #set btn pos
                            butn1.translation.set(Vec3f(butn1.translation.get().x + 0.0375,
                            butn1.translation.get().y + 0.025, butn1.translation.get().z )) #set btn pos
                            butn2.translation.set(Vec3f(butn2.translation.get().x + 0.0375,
                            butn2.translation.get().y + 0.03125, butn2.translation.get().z ))
                            #set button scale
                            butn0.scale.set(Vec3f(butn0.scale.get().x - 0.25, butn0.scale.get().y - 0.25,
                            butn0.scale.get().z - 0.25)) #set btn scale
                            butn1.scale.set(Vec3f(butn1.scale.get().x - 0.25, butn1.scale.get().y - 0.25,
                            butn1.scale.get().z - 0.25)) #set btn scale
                            butn2.scale.set(Vec3f(butn2.scale.get().x - 0.25, butn2.scale.get().y - 0.25,
                            butn2.scale.get().z - 0.25))
                            ptgeom.pointSize.set(6.0)
                    elif vpoint.fieldOfView.get()==0.0075:
                            #3rd zoom in, set FOV, button translation and scale
                            vpoint.fieldOfView.set(vpoint.fieldOfView.get() - 0.5 *
                            vpoint.fieldOfView.get()) butn0.translation.set(Vec3f(butn0.translation.get().x
                            + 0.01875, butn0.translation.get().y + 0.009375, butn0.translation.get().z ))
                            #set btn pos
                            butn1.translation.set(Vec3f(butn1.translation.get().x + 0.01875,
                            butn1.translation.get().y + 0.0125, butn1.translation.get().z )) #set btn pos
                            butn2.translation.set(Vec3f(butn2.translation.get().x + 0.01875,
                            butn2.translation.get().y + 0.015625, butn2.translation.get().z ))
                            #set button scale
                            butn0.scale.set(Vec3f(butn0.scale.get().x - 0.125, butn0.scale.get().y - 0.125,
                            butn0.scale.get().z - 0.125)) #set btn scale
                            butn1.scale.set(Vec3f(butn1.scale.get().x - 0.125, butn1.scale.get().y - 0.125,
                            butn1.scale.get().z - 0.125)) #set btn scale
                            butn2.scale.set(Vec3f(butn2.scale.get().x - 0.125, butn2.scale.get().y - 0.125,
                            butn2.scale.get().z - 0.125))
                            ptgeom.pointSize.set(12.0)
                    elif vpoint.fieldOfView.get()==0.00375:
                            #4th zoom in, set FOV, button translation and scale
                            vpoint.fieldOfView.set(vpoint.fieldOfView.get() - 0.5 *
vpoint.fieldOfView.get())
```

```
                butn0.translation.set(Vec3f(butn0.translation.get().x + 0.009375,
                butn0.translation.get().y + 0.0046875, butn0.translation.get().z )) #set btn pos
                butn1.translation.set(Vec3f(butn1.translation.get().x + 0.009375,
                butn1.translation.get().y + 0.00625, butn1.translation.get().z )) #set btn pos
                butn2.translation.set(Vec3f(butn2.translation.get().x + 0.009375,
                butn2.translation.get().y + 0.0078125, butn2.translation.get().z ))
                #set button scale
                butn0.scale.set(Vec3f(butn0.scale.get().x - 0.0625, butn0.scale.get().y - 0.0625,
                butn0.scale.get().z - 0.0625)) #set btn scale
                butn1.scale.set(Vec3f(butn1.scale.get().x - 0.0625, butn1.scale.get().y - 0.0625,
                butn1.scale.get().z - 0.0625)) #set btn scale
                butn2.scale.set(Vec3f(butn2.scale.get().x - 0.0625, butn2.scale.get().y - 0.0625,
                butn2.scale.get().z - 0.0625))
                ptgeom.pointSize.set(24.0)
        elif inputs.get() == '-':
                if vpoint.fieldOfView.get()==0.015:
                #5th (and last) zoom out
                vpoint.fieldOfView.set(vpoint.fieldOfView.get() + 1 *
vpoint.fieldOfView.get())
                butn0.translation.set(Vec3f(butn0.translation.get().x - 0.075,
                butn0.translation.get().y - 0.0375, butn0.translation.get().z ))
                butn1.translation.set(Vec3f(butn1.translation.get().x - 0.075,
                butn1.translation.get().y - 0.05, butn1.translation.get().z ))
                butn2.translation.set(Vec3f(butn2.translation.get().x - 0.075,
                butn2.translation.get().y - 0.0625, butn2.translation.get().z ))

                #set button scale
                butn0.scale.set(Vec3f(butn0.scale.get().x + 0.5005, butn0.scale.get().y +
                0.5005, butn0.scale.get().z + 0.5005))
                butn1.scale.set(Vec3f(butn1.scale.get().x + 0.5005, butn1.scale.get().y +
                0.5005, butn1.scale.get().z + 0.5005))
                butn2.scale.set(Vec3f(butn2.scale.get().x + 0.5005, butn2.scale.get().y +
                0.5005, butn2.scale.get().z + 0.5005))
                ptgeom.pointSize.set(1.5)
                elif vpoint.fieldOfView.get()==0.0075:
                #4th zoom out
                vpoint.fieldOfView.set(vpoint.fieldOfView.get() + 1 *
vpoint.fieldOfView.get())
                butn0.translation.set(Vec3f(butn0.translation.get().x - 0.0375,
                butn0.translation.get().y - 0.01875, butn0.translation.get().z ))
                butn1.translation.set(Vec3f(butn1.translation.get().x - 0.0375,
                butn1.translation.get().y - 0.025, butn1.translation.get().z ))
                butn2.translation.set(Vec3f(butn2.translation.get().x - 0.0375,
                butn2.translation.get().y - 0.03125, butn2.translation.get().z ))

                #set button scale
                butn0.scale.set(Vec3f(butn0.scale.get().x + 0.25, butn0.scale.get().y + 0.25,
                butn0.scale.get().z + 0.25))
                butn1.scale.set(Vec3f(butn1.scale.get().x + 0.25, butn1.scale.get().y + 0.25,
                butn1.scale.get().z + 0.25))
                butn2.scale.set(Vec3f(butn2.scale.get().x + 0.25, butn2.scale.get().y + 0.25,
                butn2.scale.get().z + 0.25))
                ptgeom.pointSize.set(3.0)
                elif vpoint.fieldOfView.get()==0.00375:
                #3rd zoom out                            OK
                vpoint.fieldOfView.set(vpoint.fieldOfView.get() + 1 *
vpoint.fieldOfView.get())
                butn0.translation.set(Vec3f(butn0.translation.get().x - 0.01875,
                butn0.translation.get().y - 0.009375, butn0.translation.get().z ))
```

```
                        butn1.translation.set(Vec3f(butn1.translation.get().x - 0.01875,
                        butn1.translation.get().y - 0.0125, butn1.translation.get().z ))
                        butn2.translation.set(Vec3f(butn2.translation.get().x - 0.01875,
                        butn2.translation.get().y - 0.015625, butn2.translation.get().z ))

                        #set button scale
                        butn0.scale.set(Vec3f(butn0.scale.get().x + 0.125, butn0.scale.get().y + 0.125,
                        butn0.scale.get().z + 0.125))
                        butn1.scale.set(Vec3f(butn1.scale.get().x + 0.125, butn1.scale.get().y + 0.125,
                        butn1.scale.get().z + 0.125))
                        butn2.scale.set(Vec3f(butn2.scale.get().x + 0.125, butn2.scale.get().y + 0.125,
                        butn2.scale.get().z + 0.125))
                        ptgeom.pointSize.set(6.0)
                elif vpoint.fieldOfView.get()==0.001875:
                        #2nd zoom out
                        vpoint.fieldOfView.set(vpoint.fieldOfView.get() + 1 *
vpoint.fieldOfView.get())
                        butn0.translation.set(Vec3f(butn0.translation.get().x - 0.009375,
                        butn0.translation.get().y - 0.0046875, butn0.translation.get().z ))
                        butn1.translation.set(Vec3f(butn1.translation.get().x - 0.009375,
                        butn1.translation.get().y - 0.00625, butn1.translation.get().z ))
                        butn2.translation.set(Vec3f(butn2.translation.get().x - 0.009375,
                        butn2.translation.get().y - 0.0078125, butn2.translation.get().z ))
                        #set button scale
                        butn0.scale.set(Vec3f(butn0.scale.get().x + 0.0625, butn0.scale.get().y +
                        0.0625, butn0.scale.get().z + 0.0625))
                        butn1.scale.set(Vec3f(butn1.scale.get().x + 0.0625, butn1.scale.get().y +
                        0.0625, butn1.scale.get().z + 0.0625))
                        butn2.scale.set(Vec3f(butn2.scale.get().x + 0.0625, butn2.scale.get().y +
                        0.0625, butn2.scale.get().z + 0.0625))
                        ptgeom.pointSize.set(12.0)

#instantiation of class
catcher = KeyboardCatcher()

Keyboard.characters.route(catcher)
```

#filename: MoveCarrier_noCA11.py

```
#Button Functionality of haptics device, to move carrier about screen
#accounts for TweezersManip.py: setting tweezers in/out

obj = references[0]
dis = references[1]
SwitchStylus = references[2]
cont = references[3]
px = references[4]

#haptic device button pressed/released
class MoveArrayCarrier(Dependent(SFBool)): #button either up or down
        def evaluate(self,inputs):
        #move with haptics device: button pressed
                if inputs.get() == 1: #button pressed
                        #set orientation to point away from haptic device
                        SwitchStylus.whichChoice.set(1)
                        dis.trackerPosition.route(obj.translation)
                        dis.trackerOrientation.route(obj.rotation)
                        px.point.set([  Vec3f(0, 0, 0),
                                        Vec3f(0, 0, 0.0003),
                                        Vec3f(0, 0, 0.0007),
```

```
                                        Vec3f(0, 0, 0.001),
                                        Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
                        cont.points.set([Vec3f(0, 0, 0),
                                        Vec3f(0, 0, 0.0003),
                                        Vec3f(0, 0, 0.0007),
                                        Vec3f(0, 0, 0.001),
                                        Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
                elif inputs.get() == 0: #button released
                        dis.trackerPosition.unroute(obj.translation)
                        dis.trackerOrientation.unroute(obj.rotation)
                        SwitchStylus.whichChoice.set(0)
                        px.point.set([]) #Vec3f(0,0,0)
                        cont.points.set([]) #Vec3f(0,0,0)

#instantiation of class
catcher = MoveArrayCarrier()
dis.button.route(catcher) #check for button being pressed
```

#filename: vwaif2_95.wrl

```
#VRML V2.0 utf8

DEF MYTRANS Transform {
children [
Shape {
appearance Appearance {
material
DEF _v2%0 Material {
#diffuseColor 0 0 0
#emissiveColor 0.1 0.1 0.1
transparency 1
}
surface DEF SSURF FrictionalSurface {
#stiffness 100 #default is 900 degree of repulsive normal force
#damping 0.02 #default is 0 damping of repulsive normal force; energy absorbing quantity
startingFriction 0.0605 #measured
stoppingFriction 0.0185 #estimated
dynamicFriction 0.0395 #measured
stiffness 441 #default is 900 N/m; documented
stiffnessT 200 #estimated
damping 0.01 #default is 0; documented
} #simple surface for now: change; to force model
}
geometry
IndexedFaceSet {
```

```
coord
Coordinate {
point [ -7.8611 0.83622 1.8832,
-8.3581 0.37591 2.0021,
-8.0641 -0.00018916 2.0108,
-6.7558 1.0027 1.7671,
-6.3669 0.84951 1.849,
-7.1862 0.53593 1.9798,
-7.8701 0.40178 1.988,
-7.2241 0.21537 2.0736,
-5.9337 0.74053 1.9696 ]
}
solid FALSE
convex FALSE
creaseAngle 0.5
coordIndex [ 7, 5, 6, -1, 3, 5, 4, -1,
6, 0, 1, -1, 0, 6, 5, -1,
3, 0, 5, -1, 7, 6, 2, -1,
8, 5, 7, -1, 5, 8, 4, -1,
6, 1, 2, -1 ]
}
}
Shape {
appearance
Appearance {
material USE _v2%0
 surface USE SSURF}
geometry
IndexedFaceSet {
coord
Coordinate {
point [

…

]}}]}
DEF ROTATECOCHLEA PythonScript {
url "RotateBoneKeyR1.py"
references [USE MYTRANS]
}
```

#filename: ForceDataCapture29f.py

```
forcetorque = references[0]
dis = references[1]
cont = references[2]
px = references[3]

class DataLogger( TypedField(Dependent(SFString), Dependent(SFVec3f) ) ):
	def evaluate( self, inputs ):
		v = inputs[0].get() #position
		data_file = open('data_logging_all.txt', 'a') #open text file: w will truncate
		data_file.write('%.8f' % (dis.devicePosition.get().x) + ',' + '%.8f' % (dis.devicePosition.get().y)
		+ ',' '%.8f' % (dis.devicePosition.get().z) + ',')
		#add accumulative effect of frictional force: only interested in z movement
		#if device position is less than proxy position: moving out of ST; increase Ff
		f_z = forcetorque.force.get().z
		f_y = forcetorque.force.get().y
		f_x = forcetorque.force.get().x
```

```
if f_x < 0:
	f_xabs = -1*f_x
else:
	f_xabs = f_x
if dis.devicePosition.get().z < dis.proxyPosition.get().z and forcetorque.force.get().z != 0:
	forcetorque.force.set(Vec3f(f_x, f_y, f_z + f_xabs*0.0395))
#elif device position is greater than proxy position: moving into ST; reduce Ff
elif dis.devicePosition.get().z > dis.proxyPosition.get().z and forcetorque.force.get().z != 0:
	forcetorque.force.set(Vec3f(f_x, f_y, f_z - f_xabs*0.0395))
#elif device position is equal to proxy position: keep Ff same.
if forcetorque.force.get().z != 0 and dis.devicePosition.get().z >= -3.4:
	#define ORIGINAL point set
	px.point.set([	Vec3f(0, 0, 0),
				Vec3f(0, 0, 0.0003),
				Vec3f(0, 0, 0.0007),
				Vec3f(0, 0, 0.001),
				Vec3f(0, 0, 0.0015),
				Vec3f(0, 0, 0.0035),
				Vec3f(0, 0, 0.005),
				Vec3f(0, 0, 0.00695),
				Vec3f(0, 0, 0.008),
				Vec3f(0, 0, 0.01),
				Vec3f(0, 0, 0.012),
				Vec3f(0, 0, 0.014)])
	cont.points.set([Vec3f(0, 0, 0),
				Vec3f(0, 0, 0.0003),
				Vec3f(0, 0, 0.0007),
				Vec3f(0, 0, 0.001),
				Vec3f(0, 0, 0.0015),
				Vec3f(0, 0, 0.0035),
				Vec3f(0, 0, 0.005),
				Vec3f(0, 0, 0.00695),
				Vec3f(0, 0, 0.008),
				Vec3f(0, 0, 0.01),
				Vec3f(0, 0, 0.012),
				Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -3.4 and
dis.devicePosition.get().z >= -4:
	px.point.set([	Vec3f(0, 0, 0.0003),
				Vec3f(0, 0, 0.0007),
				Vec3f(0, 0, 0.001),
				Vec3f(0, 0, 0.0015),
				Vec3f(0, 0, 0.0018),
				Vec3f(0, 0, 0.0035),
				Vec3f(0, 0, 0.005),
				Vec3f(0, 0, 0.00695),
				Vec3f(0, 0, 0.008),
				Vec3f(0, 0, 0.01),
				Vec3f(0, 0, 0.012),
				Vec3f(0, 0, 0.014)])
	cont.points.set([Vec3f(0, 0, 0.0003),
				Vec3f(0, 0, 0.0007),
				Vec3f(0, 0, 0.001),
				Vec3f(0, 0, 0.0015),
				Vec3f(0, 0, 0.0018),
				Vec3f(0, 0, 0.0035),
				Vec3f(0, 0, 0.005),
				Vec3f(0, 0, 0.00695),
				Vec3f(0, 0, 0.008),
				Vec3f(0, 0, 0.01),
```

```
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -4 and
dis.devicePosition.get().z >= -5:
        px.point.set([      Vec3f(0, 0, 0.001),
                                        Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0018),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
        cont.points.set([Vec3f(0, 0, 0.001),
                                        Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0018),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                                Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -5 and
dis.devicePosition.get().z >= -8:
        px.point.set([      Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0018),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
        cont.points.set([Vec3f(0, 0, 0.0015),
                                        Vec3f(0, 0, 0.0018),
                                        Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -8 and
dis.devicePosition.get().z >= -9:
        px.point.set([      Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
```

```
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.009),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.011),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
        cont.points.set([Vec3f(0, 0, 0.0035),
                                        Vec3f(0, 0, 0.0038),
                                        Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.009),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.011),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -9 and
dis.devicePosition.get().z >= -10:
        px.point.set([Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.085),
                                        Vec3f(0, 0, 0.009),
                                        Vec3f(0, 0, 0.0095),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.011),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
        cont.points.set([Vec3f(0, 0, 0.005),
                                        Vec3f(0, 0, 0.0058),
                                        Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.085),
                                        Vec3f(0, 0, 0.009),
                                        Vec3f(0, 0, 0.0095),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.011),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -10 and
dis.devicePosition.get().z >= -11:
        px.point.set([      Vec3f(0, 0, 0.00695),
                                        Vec3f(0, 0, 0.0075),
                                        Vec3f(0, 0, 0.008),
                                        Vec3f(0, 0, 0.085),
                                        Vec3f(0, 0, 0.009),
                                        Vec3f(0, 0, 0.0095),
                                        Vec3f(0, 0, 0.01),
                                        Vec3f(0, 0, 0.0105),
                                        Vec3f(0, 0, 0.011),
                                        Vec3f(0, 0, 0.0115),
                                        Vec3f(0, 0, 0.012),
                                        Vec3f(0, 0, 0.014)])
```

```
        cont.points.set([Vec3f(0, 0, 0.00695),
                                    Vec3f(0, 0, 0.0075),
                                    Vec3f(0, 0, 0.008),
                                    Vec3f(0, 0, 0.085),
                                    Vec3f(0, 0, 0.009),
                                    Vec3f(0, 0, 0.0095),
                                    Vec3f(0, 0, 0.01),
                                    Vec3f(0, 0, 0.0105),
                                    Vec3f(0, 0, 0.011),
                                    Vec3f(0, 0, 0.0115),
                                    Vec3f(0, 0, 0.012),
                                    Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -11 and
dis.devicePosition.get().z >= -13:
        px.point.set([      Vec3f(0, 0, 0.008),
                                    Vec3f(0, 0, 0.085),
                                    Vec3f(0, 0, 0.009),
                                    Vec3f(0, 0, 0.0095),
                                    Vec3f(0, 0, 0.01),
                                    Vec3f(0, 0, 0.0105),
                                    Vec3f(0, 0, 0.011),
                                    Vec3f(0, 0, 0.0115),
                                    Vec3f(0, 0, 0.012),
                                    Vec3f(0, 0, 0.0125),
                                    Vec3f(0, 0, 0.013),
                                    Vec3f(0, 0, 0.014)])
        cont.points.set([   Vec3f(0, 0, 0.008),
                                    Vec3f(0, 0, 0.085),
                                    Vec3f(0, 0, 0.009),
                                    Vec3f(0, 0, 0.0095),
                                    Vec3f(0, 0, 0.01),
                                    Vec3f(0, 0, 0.0105),
                                    Vec3f(0, 0, 0.011),
                                    Vec3f(0, 0, 0.0115),
                                    Vec3f(0, 0, 0.012),
                                    Vec3f(0, 0, 0.0125),
                                    Vec3f(0, 0, 0.013),
                                    Vec3f(0, 0, 0.014)])
elif forcetorque.force.get().z != 0 and dis.devicePosition.get().z < -13:
        px.point.set([      Vec3f(0, 0, 0.01),
                                    Vec3f(0, 0, 0.0105),
                                    Vec3f(0, 0, 0.011),
                                    Vec3f(0, 0, 0.0115),
                                    Vec3f(0, 0, 0.01175),
                                    Vec3f(0, 0, 0.012),
                                    Vec3f(0, 0, 0.0125),
                                    Vec3f(0, 0, 0.01275),
                                    Vec3f(0, 0, 0.013),
                                    Vec3f(0, 0, 0.0135),
                                    Vec3f(0, 0, 0.01375),
                                    Vec3f(0, 0, 0.014)])
        cont.points.set([   Vec3f(0, 0, 0.01),
                                    Vec3f(0, 0, 0.0105),
                                    Vec3f(0, 0, 0.011),
                                    Vec3f(0, 0, 0.0115),
                                    Vec3f(0, 0, 0.01175),
                                    Vec3f(0, 0, 0.012),
                                    Vec3f(0, 0, 0.0125),
                                    Vec3f(0, 0, 0.01275),
                                    Vec3f(0, 0, 0.013),
```

```
                                        Vec3f(0, 0, 0.0135),
                                        Vec3f(0, 0, 0.01375),
                                        Vec3f(0, 0, 0.014)])
        data_file.write('%.8f' % (forcetorque.force.get().x) + ',' + '%.8f' % (forcetorque.force.get().y) + ','
        '%.8f' % (forcetorque.force.get().z) + ',')
        print >> data_file, '%.8f' % (forcetorque.torque.get().x) + ',' + '%.8f' %
        (forcetorque.torque.get().y) + ',' '%.8f' % (forcetorque.torque.get().z) + ','
        data_file.close() #close text file
        return '%.3f %.3f %.3f' % ( v.x, v.y, v.z ) #return for buttons on GUI

pos_string = DataLogger()
force_string = DataLogger()
torque_string = DataLogger()
```

Appendix F

Make file

#filename: Makefile

```
#
# Copyright (c) 1997-2002, Reachin Technologies AB.
# ALL RIGHTS RESERVED
#
# Borland C++ Builder Makefile
#
#
# This is a makefile for producing binaries that use the Reachin API DLL.
# The file produces two basic targets; an optimised or debugged EXE.
# The object files and target will be placed into either an "opt" or
# "dbg" subdirectory respectively.
#
#
# To compile the entire EXE in optimised mode, you can execute:
#     make opt
# And to compile an individual object file in optimised mode:
#     make opt\Tool.obj
#
# To compile the entire EXE in debugged mode, you can execute:
#     make dbg
# And to compile an individual object file in debugged mode:
#     make -DDEBUG dbg\Tool.obj
#


.AUTODEPEND

# The file Settings.bcb (by default located in "..\Settings.bcb")
# should be used to customise your insallation of the API and the
# components required by the API (e.g. the location of ZLib and
# libpng).
# The file Rules.bcb contains the compiler flags used to produce a
# DLL or executable that uses the API. It is strongly recommended that
# you do not change the contents of the Rules file.

!if $d(REACHIN_ROOT)
!include $(REACHIN_ROOT)\Settings.bcb
!include $(REACHIN_ROOT)\Rules.bcb
!else
!include "..\Settings.bcb"
!include "..\Rules.bcb"
!endif


# Define the EXE target for this Makefile:
TARGET=Cochlear_node


# add any user flags, user -Ddefines, etc, here.
# CUSERFLAGS added to a C++ compile line,
# LUSERFLAGS added to a link line.
CUSERFLAGS=
```

```
LUSERFLAGS=

# Specify -DVERBOSE on the make command line (note: must come before
# the target to be built, eg make -DVERBOSE opt).
!if $d(VERBOSE)
CC=bcc32
!else
CC=@bcc32
!endif

# Define your C++ classes here. Each entry should be the name for
# an object file that will be produced and linked into the target
# DLL or executable. Note: You must make sure that the last entry
# does not have any white space after it!


CLASSES =               \
        CochlearV \
        CochlearBuilder \
        STympV \
        STympH \
        BMembV \
        Stylet \
        CArr\
        $(NULL)


# Convert the above list into a set of make targets to build
OBJSEXTS=$(CLASSES:^   =.obj)
DIRSLASH=$(DIR)\\
OBJTMP=$(OBJSEXTS:^ =$(DIRSLASH))
OBJS=$(DIR)\$(OBJTMP:.obj=.obj )

# By default we build the target EXE.
default: $(DIR)\$(TARGET).exe

# This will install the current target into the resources directory.
# It will not maintain separate folders for dbg or opt binaries; the
# user is expected to track this.
install: $(DIR)\$(TARGET).exe
        -mkdir $(REACHIN_ROOT)\bin\$(DIR)
        @copy $(DIR)\$(TARGET).exe $(REACHIN_ROOT)\bin\$(DIR)
        @echo Installed $(DIR) $(TARGET) in $(REACHIN_ROOT)\bin\$(DIR).
        @copy $(DIR)\$(TARGET).exe $(REACHIN_ROOT)\bin
        @echo Installed $(DIR) $(TARGET) in $(REACHIN_ROOT)\bin.

$(DIR)\$(TARGET).exe: $(DIR)\.marker $(OBJS)
        $(CC) -e$@ $(EXE_LINKF) -L$(REACHIN_ROOT)\lib\nt \
        -L$(REACHIN_ROOT)\lib\$(DIR) \
        $(OBJS) ReachinAPI.lib ReachinAPIExtras.lib $(LDLIBS)
        copy $(DIR)\$(TARGET).exe $(DIR)\..
```

Appendix G

Cochlear Builder

#filename: CochlearBuilder.cpp

```
#include "Vrml.h"
#include "Scene.h"

using namespace Reachin;

void main( const int argc,
                        const char *argv[] ) {

 if( argc != 2 ) {
        cerr << "Usage: " << argv[0] << " url" << endl;
        exit(1);
 }

 try {
        Group *g = VRML::createVrmlFromURL( argv[1] );

        // start the simulation
        Scene::startScene();
 }
 catch( Error::QuitAPI ) {
 }
 catch( Error::ErrorBase &e ) {
        cerr << e << endl;
 }
}
```

Appendix H

Carrier Movement

#filename: MoveCarrierR4AB.py

```
#Button Functionality of haptics device, to move carrier about screen
#accounts for TweezersManip.py: setting tweezers in/out

#Mov1 = references[0]
dis = references[1] #display node

#haptic device button pressed/released
class MoveArrayCarrier(Dependent(SFBool)): #button either up or down
	def evaluate(self,inputs):
	#move with haptics device: button pressed
		if inputs.get() == 1: #button pressed
			#set orientation to point away from haptic device
			dis.trackerPosition.route(references[0].translation)
			dis.trackerOrientation.route(references[0].rotation)
		elif inputs.get() == 0: #button released
			dis.trackerPosition.unroute(references[0].translation)
			dis.trackerOrientation.unroute(references[0].rotation)

#instantiation of class
catcher = MoveArrayCarrier()
dis.button.route(catcher) #check for button being pressed
```

Appendix I

Stylet Withdrawal

#filename: Cochlear_colcont.wrl

```
Import {
url [
"urn:inet:reachin.se:/display/tools/toolsCI/fOUT_comb.wrl",
"urn:inet:reachin.se:/display/tools/toolsCI/fIN_comb.wrl",
"urn:inet:reachin.se:/display/tools/toolsCI/hook_comb.wrl",
"urn:inet:reachin.se:/display/sceneCI/centre_loop_2.wrl"
]}

DEF CON CollidingController {
mass 0.1
proxyRadius 0.004
springStiffness 600
rotSpringStiffness 0.1
inertia BoxInertia {
size 0.03 0.02 0.02
mass 0.1
}
children [
Shape {
appearance Appearance{ }
geometry IndexedLineSet {
coord DEF PX_COORDS Coordinate {
point [    0.005 0 0, 0.0075 0 0,
0.0075 0 0, 0.01 0 0,
0.01 0 0, 0.01 0.0025 0,
0.01 0.0025 0, 0.01 0.005 0
]} #end coordinate
coordIndex [0 1, 2 3, 4 5, 6 7]
} #end geometry
} #end shape
]#end children of colliding controller
points [   0.005 0 0, 0.0075 0 0,
0.0075 0 0, 0.01 0 0,
0.01 0 0, 0.01 0.0025 0,
0.01 0.0025 0, 0.01 0.005 0
]
collisionChild DEF CG Group {
#chose hook collision scene
children [
#loop
USE LOOP_TRANS #have button simple surface nodes x 3 in wrl file centre_loop_2.wrl
]#end children of collisionChild
}}#end CON

DEF CYL_TIP Transform {
children [
Shape {
#cylinder as tip
appearance Appearance{
material Material {diffuseColor 1 0 0}
}
geometry Cylinder{
```

```
radius       0.0003
height       0.005
side         TRUE
bottom       TRUE
top          TRUE
}  #end geometry cylinder
} #end shape
]} #end transform

DEF BOX_LENGTH Transform {
#  translation 0 -0.02 0 #check, adjust
translation 0 0 0.024
children [
Shape {
#cylinder as length also
appearance Appearance{
material Material {diffuseColor 0 1 0}
}
geometry Box{   #for now
size 0.001 0.001 0.04
}  #end geometry cylinder
} #end shape
]} #end transform

DEF DIS Display {

proxyRadius 0.004 #haptics for tip cylinder: may change later

tip Shape{} #both shape nodes
stylus USE BOX_LENGTH

children [
USE CON
USE fOUT_comb, USE hook_comb, USE LINE #show available tool selection
Cochlear {
cont [
DEF MOVECA CArr {
scale          0.009 0.009 0.009
}]
styl [
DEF MOVES Stylet {
scale          0.009 0.009 0.009
}]} #end cochlear
] #end children
#stylus Transform {
#children [
#DEF SWSTYL Switch {
#whichChoice 0
#choice [
#USE FOUT
#USE FIN #only selectable with button
#] #end choice
#} #end switch
#]
#} #end stylus

#tip Transform {
#children [
#DEF SWTIP Switch {
#whichChoice 0
```

```
#choice [
#USE FOUT_T
#USE FIN_T #only selectable with button
#USE NO_TIP
#] #end choice
#} #end switch
#]
#} #end tip

}#end display

##change stylus/tip combination (keyboard - f/h)
#DEF SELECTTOOL PythonScript {
# url "SelectToolR4.py"
# references [ USE PX_COORDS, USE SWSTYL, USE SWTIP, USE DIS, USE CON, USE SWCON ]
#}

##move carrier (button)
DEF MOVECARRIER PythonScript {
url "MoveCarrierR5.py"
references [ USE MOVECA, USE DIS, USE LOOP_TRANS ]
}

##move stylet (button)
DEF MOVECARRIER PythonScript {
url "MoveCarrierR5.py"
references [ USE MOVES, USE DIS, USE LOOP_TRANS ]
}

# Python script node
DEF touchHandler_psct PythonScript {
url "LoopTouchedR4.py"
references [ USE MOVES, USE LINNER_T ]
}

ROUTE SURF1.armed TO touchHandler_psct.loop1Touched
ROUTE touchHandler_psct.loop1Touched TO _v2%0.diffuseColor
ROUTE SURF2.armed TO touchHandler_psct.loop2Touched
ROUTE touchHandler_psct.loop2Touched TO _v2%0.diffuseColor
ROUTE SURF3.armed TO touchHandler_psct.loop3Touched
ROUTE touchHandler_psct.loop3Touched TO _v2%0.diffuseColor
ROUTE CON.pointProxies TO PX_COORDS.point
```

#filename: centre_loop_2.wrl

```
DEF LOOP_TRANS Transform {
#translation 0.04 0.00 0.00
scale          0.009 0.009 0.009
children [
Transform {
rotation 0 1 0 1.5705
children [
Transform {
translation 36.003 -0.1 0
children [
Shape {
appearance
Appearance {
```

```
material
DEF _v2%0 Material {
diffuseColor        0 0 0
emissiveColor       0.1 0.1 0.1
}
surface DEF SURF1 ButtonSimpleSurface {}
}
geometry
IndexedLineSet {
coord
Coordinate {
point       [ -32.685 -0.745 4e-009,
-32.685 -0.75258 4e-009,
-32.685 -0.75258 4e-009,

...

-31.826 -0.82683 4e-009,
-31.824 -0.81654 4e-009 ]
}

colorPerVertex      FALSE
coordIndex          [ 0, 1, -1, 2, 3, -1, 4, 5,
-1, 6, 7, -1, 8, 9, -1, 10,

...

192, 193, -1, 194, 195, -1, 196, 197,
-1, 198, 199, -1 ]
}},
Shape {
appearance
Appearance {
material USE _v2%0
surface DEF SURF2 ButtonSimpleSurface {}
}
geometry
IndexedLineSet {
coord
Coordinate {
point       [ -31.824 -0.81654 4e-009,
-31.821 -0.80625 4e-009,

...

-31.864 -0.57164 4e-009,
-31.859 -0.58162 4e-009 ]
}

colorPerVertex      FALSE
coordIndex          [ 0, 1, -1, 2, 3, -1, 4, 5,
-1, 6, 7, -1, 8, 9, -1, 10,

...

192, 193, -1, 194, 195, -1, 196, 197,
-1, 198, 199, -1 ]
}},
Shape {
appearance
```

```
Appearance {
material USE _v2%0
surface DEF SURF3 ButtonSimpleSurface {}
}
geometry
IndexedLineSet {
coord
Coordinate {
point      [ -31.859 -0.58162 4e-009,
-31.854 -0.59167 4e-009,
...
-31.815 -0.74248 4e-009,
-31.815 -0.745 4e-009 ]
}

colorPerVertex    FALSE
coordIndex        [ 0, 1, -1, 2, 3, -1, 4, 5,
-1, 6, 7, -1, 8, 9, -1, 10,

...

27, -1, 28, 29, -1, 30, 31, -1,
32, 33, -1, 34, 35, -1 ]
}}]}]}]}
```

#filename: LoopTouchedR4.py

```
#creating field getting and setting indirectly via dm.findField,
#for styl.inner_trans field, exposed in Stylet.cpp

from PythonScript import *

styl = references[0]
loop = references[1]

nodes, dm = createVrmlFromURL("StyletOnlyA.wrl") #dm made public

class touchHandlerClass( TypedField( SFColor , None , SFBool ) ):
        def __init__( self, inColor ):
                SFColor.__init__( self )
                self.originalColor = inColor
                self.hit = 1
        def evaluate( self , inputs ):
                if self.hit == 1:
                        self.hit = 0
                        #move stylet
                        a = dm.findField("INNER_T.translation").get() #value of translation field
                        mv_trans = Vec3f(-1,0,0)
                        dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
                        new_f = dm.findField("INNER_T.translation").get()
                        styl.inner_trans.set(a + mv_trans) #set(new_f)
                        styl.inner_trans.touch() #set as new value
                        #move loop
                        loop.translation.set(a + mv_trans)
                        return Color ( 0 , 0 , 1 )
                else:
                        self.hit = 1
                        return self.originalColor
```

```
                    #don't move stylet and loop: button not pressed

loop2Touched   = touchHandlerClass( Color(1,0,0) )
loop1Touched = touchHandlerClass( Color(0,1,0) )
loop3Touched = touchHandlerClass( Color(0,0,1) )
```

#filename: SelectToolR4.py

```
#Select tool for use in cochlear implant surgical simulation

import Keyboard

px = references[0]
SwitchStylus = references[1]
SwitchTip = references[2]
dis = references[3]
cont = references[4]
sc = references[5]

class KeyboardCatcher(Dependent(SFString)):
        def evaluate(self,inputs):
        #FORCEPS OPEN - DEFAULT
                if inputs.get() == 'h':
                        SwitchStylus.whichChoice.set(2)
                        SwitchTip.whichChoice.set(2)
                        sc.whichChoice.set(0)
                        cont.pointProxies.route(px.point)
                        dis.proxyRadius.set(0.00005)
                elif inputs.get() == 'f':
                        SwitchTip.whichChoice.set(0)
                        SwitchStylus.whichChoice.set(0)
                        dis.proxyRadius.set(0.0005)
                        sc.whichChoice.set(-1)
                        cont.pointProxies.unroute(px.point)

#instantiation of class
catcher = KeyboardCatcher()

Keyboard.characters.route(catcher)
```

#filename: Cochlear_ftgNoLoop.wrl

```
Import {
url [
"urn:inet:reachin.se:/display/tools/toolsCI/fOUT_comb.wrl",
"urn:inet:reachin.se:/display/tools/toolsCI/fIN_comb.wrl",
"urn:inet:reachin.se:/display/tools/toolsCI/hook_comb.wrl",
"urn:inet:reachin.se:/bin/UserInterface"
  ]
}

DEF CYL_TIP Transform {
children [
Shape {
#cylinder as tip
appearance Appearance{
material Material {diffuseColor 1 0 0}
}
geometry Cylinder{
```

```
radius       0.0003
height       0.005
side         TRUE
bottom       TRUE
top          TRUE
}  #end geometry cylinder
} #end shape
]} #end transform

DEF BOX_LENGTH Transform {
translation 0 0 0.024
children [
Shape {
#cylinder as length also
appearance Appearance{
material Material {diffuseColor 0 1 0}
}
geometry Box{   #for now
size 0.001 0.001 0.04
}  #end geometry cylinder
} #end shape
]} #end transform

DEF DIS Display {
proxyRadius 0.0005 #0.005 too big - RUNAWAY(1) error generated

tip USE CYL_TIP #Shape{} #both shape nodes
stylus USE BOX_LENGTH

children [
Transform {
translation -0.1 -0.1 0
children [
Frame {
children [
DEF POS_BUTTON Button {
desiredDimensions 0.05 0.01 0.001
}
DEF FORCE_BUTTON Button {
desiredDimensions 0.05 0.01 0.001
}]}]}

USE fOUT_comb, USE hook_comb, USE LINE #show available tool selection

Cochlear {
cont [
DEF MOVECA CArr {
scale          0.009 0.009 0.009
}]
styl [
DEF FTG ForceTorqueGroup {
children [
DEF MOVES Stylet {
scale          0.009 0.009 0.009
}]
}#end forcetorquegroup
] } #end cochlear
] #end children

} #end display
```

```
##move carrier (button)
DEF MOVECARRIER PythonScript {
url "MoveCarrierR6.py"
references [ USE MOVECA, USE DIS ] #, USE SWSTYL, USE LINE, USE SWCON ]
}

##move stylet (button)
DEF MOVECARRIER PythonScript {
url "MoveCarrierR6.py"
references [ USE MOVES, USE DIS ] #, USE SWSTYL, USE LINE, USE SWCON ]
}

DEF PS PythonScript {
url "ForceT.py"
references [ USE MOVES ]
}

# set up routes for the button text

ROUTE DIS.trackerPosition TO PS.pos_string
ROUTE PS.pos_string TO POS_BUTTON.text
ROUTE FTG.force TO PS.force_string
ROUTE PS.force_string TO FORCE_BUTTON.text
```

#filename: ForceT.py

```
from PythonScript import *

styl = references[0]
nodes, dm = createVrmlFromURL("StyletOnlyA.wrl") #dm made public

class Vec3fToString( TypedField( SFString, SFVec3f ) ):
    def evaluate( self, inputs ):
        v = inputs[0].get()
        if (v.x * v.x) > 0.05: #was 0.05
        #force in x-direction > 0, so move stylet
        #current value of translation field for stylet, should change
        a = dm.findField("INNER_T.translation").get()
            mv_trans = Vec3f(-0.1,0,0)
            dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
            new_f = dm.findField("INNER_T.translation").get()
            styl.inner_trans.set(a + mv_trans) #set(new_f)
            styl.inner_trans.touch()
        elif (v.y * v.y) > 0.05: #was 0.05
            #force in x-direction > 0, so move stylet
            a = dm.findField("INNER_T.translation").get()
            #current value of translation field for stylet, should change
            mv_trans = Vec3f(-0.1,0,0)
            dm.findField("INNER_T.translation").set(a + mv_trans)
            new_f = dm.findField("INNER_T.translation").get()
            styl.inner_trans.set(a + mv_trans) #set(new_f)
            styl.inner_trans.touch()
        elif (v.z * v.z) > 0.05: #was 0.05
            #force in x-direction > 0, so move stylet
            a = dm.findField("INNER_T.translation").get()
            #current value of translation field for stylet, should change
            mv_trans = Vec3f(-0.1,0,0)
            dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
```

```
                new_f = dm.findField("INNER_T.translation").get()
                styl.inner_trans.set(a + mv_trans) #set(new_f)
                styl.inner_trans.touch()
        return '%.2f %.2f %.2f' % ( v.x, v.y, v.z )

force_string = Vec3fToString()
pos_string = Vec3fToString()
```

#filename: KeysR3.wrl

```
#Select tool for use in cochlear implant surgical simulation
#includes zoom in and out capabilities
#also includes withdrawal of stylet using key 'w',
#instead of user-induced force application

from PythonScript import *

import Keyboard

SwitchStylus = references[1]
SwitchTip = references[2]
dis = references[0]
tmp = references[3]
car = references[4]
styl = references[5] #MOVESV, visual representation of stylet
mem = references[6]
fout = references[7]
fin = references[8]
tip = references[9]
len = references[10]

nodes, dm = createVrmlFromURL("StyletOnlyA.wrl") #dm made public

class KeyboardCatcher(Dependent(SFString)):
        def evaluate(self,inputs):
        #FORCEPS OPEN - DEFAULT
                if inputs.get() == 'h':
                        SwitchStylus.whichChoice.set(2)
                        SwitchTip.whichChoice.set(1)
                        dis.proxyRadius.set(0.000015) #was 0.00005, 0.0000025
                elif inputs.get() == 'f':
                        SwitchTip.whichChoice.set(0)
                        SwitchStylus.whichChoice.set(0)
                        dis.proxyRadius.set(0.000025) #was 0.0005
                elif inputs.get() == 'r':
                        #partial withdrawal of stylet
                        a = dm.findField("INNER_T.translation").get()
                        mv_trans = Vec3f(-6,0,0)
                        dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
                        new_f = dm.findField("INNER_T.translation").get()
                        styl.inner_trans.set(a + mv_trans) #set(new_f)
                        styl.inner_trans.touch()
                        if styl.inner_trans.get().x < -31:
                                #make stylet 'disappear' from scene
                                #styl.scale.set(Vec3f(0,0,0)) #set scale to 0 - doesn't work
                                styl.inner_trans.set(Vec3f(-10000, 0, 0)) #push way off screen
                elif inputs.get() == 'e':
                        while styl.inner_trans.get().x > -31:
                                a = dm.findField("INNER_T.translation").get()
                                mv_trans = Vec3f(-0.003,0,0)
                                dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
```

```
                    new_f = dm.findField("INNER_T.translation").get()
                    styl.inner_trans.set(a + mv_trans) #set(new_f)
                    styl.inner_trans.touch()
            #return styl.inner_trans.get().x
            if styl.inner_trans.get().x < -31:
                    #make stylet 'disappear' from scene
                    styl.inner_trans.set(Vec3f(-10000, 0, 0))
            elif inputs.get() == 'w':
            #withdraw stylet, in incremental steps
            #current value of translation field for stylet, should change
            a = dm.findField("INNER_T.translation").get()
            mv_trans = Vec3f(-0.3,0,0) #was Vec3f(-0.1,0,0), make withdrawal faster
            dm.findField("INNER_T.translation").set(a + mv_trans) #self.inc
            new_f = dm.findField("INNER_T.translation").get()
            styl.inner_trans.set(a + mv_trans) #set(new_f)
            styl.inner_trans.touch()
            if styl.inner_trans.get().x < -31:
                    #make stylet 'disappear' from scene
                    #styl.scale.set(Vec3f(0,0,0)) #set scale to 0 - doesn't work
                    styl.inner_trans.set(Vec3f(-10000, 0, 0)) #push way off screen

#instantiation of class
catcher = KeyboardCatcher()

Keyboard.characters.route(catcher)
```

Appendix J

Spline Derivation

#filename: Acquire Final Stats Sim.xls

```
Sub SubSampleSplinePts()
'this subroutine is designed to sub-sample the data at specific points along the insertion distance,
'then interpolate the value at this point, based on the two surrounding points.

Dim i As Integer, Cell3 As Integer
Dim x1 As Double, x2 As Double, ni As Double, F1 As Double, F2 As Double, Fi As Double, m As
Double, x As Double, a As Double

ni = 0 'initial condition for distance value required
Cell3 = 0

With Range("A1")
    NCellDown = Range(.Offset(1, 0), .End(xlDown)).Rows.Count 'number of rows (doesn't change)

'MsgBox "the number of rows is: " & NCellDown
For i = 1 To NCellDown
        'MsgBox "the number of iterations is:" & i
        x1 = .Offset(i - 1, 0).Cells.Value
        'MsgBox "x1 is: " & x1
        x2 = .Offset(i, 0).Cells.Value
        'MsgBox "x2 is: " & x2
        F1 = .Offset(i - 1, 1).Cells.Value
        'MsgBox "F1 is: " & F1
        F2 = .Offset(i, 1).Cells.Value
        'MsgBox "F2 is: " & F2
        a = 1
        Do While x1 <= ni And ni <= x2 And a > 0
        If x1 < ni And ni < x2 Then
                'the value lies between the two points
                'determine the value of the force between these two points
                m = (F2 - F1) / (x2 - x1)
                x = Abs(ni - x1)
                Fi = F1 + m * x
                'write the distance and force values to the next cell
                .Offset(Cell3, 3) = ni
                .Offset(Cell3, 4) = Fi
                Cell3 = Cell3 + 1 'increment cell to write in the distance and force values
                ni = ni + 0.1 'increment distance value to be determined
                'MsgBox "in this loop 1"
        ElseIf x1 = ni Then
                Fi = F1
                'write the distance and force values to the next cell
                .Offset(Cell3, 3) = ni
                .Offset(Cell3, 4) = Fi
                Cell3 = Cell3 + 1 'increment cell to write in the distance and force values
                ni = ni + 0.1
                'MsgBox "in this loop 2"
        ElseIf x2 = ni Then
                Fi = F2
                'write the distance and force values to the next cell
```

```
                .Offset(Cell3, 3) = ni
                .Offset(Cell3, 4) = Fi
                Cell3 = Cell3 + 1 'increment cell to write in the distance and force values
                ni = ni + 0.1
                'MsgBox "in this loop 3"
        Else
                'leave...check next cell: not within region
                a = -1
        End If
Loop
Next i
End With

End Sub
```

Appendix K

Spline Function Coefficients

#filename: Acquire Final Stats Sim.xls

Rank 318 Eqn 6849 Fourier Series Polynomial 9x2

r^2 Coef Det DF Adj r^2 Fit Std Err F-value
0.9999984931 0.9999984011 5.56084e-05 1.150286e+07

Parm	Value	Std Error	t-value	95% Confidence Limits		P>|t|
a	-36.2703633	3.012120090	-12.0414732	-42.1970002	-30.3437263	0.00000
b	-6.44922322	0.217046144	-29.7136042	-6.87628246	-6.02216399	0.00000
c	67.87966114	5.604995000	12.11056587	56.85129261	78.90802967	0.00000
d	55.27219297	4.505290498	12.26828614	46.40759916	64.13678677	0.00000
e	10.41022081	0.350642514	29.68898632	9.720297828	11.10014379	0.00000
f	11.02762991	0.365990179	30.13094489	10.30750891	11.74775090	0.00000
g	-38.9915435	3.108844418	-12.5421341	-45.1084949	-32.8745921	0.00000
h	-23.6026455	1.820267758	-12.9665789	-27.1841980	-20.0210930	0.00000
i	-8.90142687	0.289210916	-30.7783226	-9.47047725	-8.33237648	0.00000
j	-5.68316640	0.179095582	-31.7325885	-6.03555424	-5.33077856	0.00000
k	12.05458633	0.887120720	13.58843962	10.30909071	13.80008196	0.00000
l	5.055726206	0.348760486	14.49627011	4.369506295	5.741946116	0.00000
m	2.844011034	0.086316704	32.94855913	2.674174589	3.013847479	0.00000
n	1.083367663	0.031109808	34.82399092	1.022156115	1.144579212	0.00000
o	-1.65455761	0.104808816	-15.7864354	-1.86077907	-1.44833615	0.00000
p	-0.38755342	0.021685382	-17.8716438	-0.43022151	-0.34488534	0.00000
q	-0.28553526	0.007625421	-37.4451791	-0.30053901	-0.27053150	0.00000
r	-0.04218204	0.000979867	-43.0487507	-0.04411002	-0.04025406	0.00000
s	0.050734522	0.002359420	21.50296624	0.046092136	0.055376908	0.00000

Area Xmin-Xmax Area Precision
0.9381915723 1.369024e-09

Function min X-Value Function max X-Value
0.0037828809 1.009197e-10 0.1344002281 15.932616745

1st Deriv min X-Value 1st Deriv max X-Value
-0.011072019 16.500000000 0.0270218993 8.5459404638

2nd Deriv min X-Value 2nd Deriv max X-Value
-0.001912733 0.8250000000 0.0212722017 7.3149284884

Soln Vector Covar Matrix
GaussElim LUDecomp
r^2 Coef Det DF Adj r^2 Fit Std Err Max Abs Err
0.9999984931 0.9999984011 5.56084e-05 0.0001569002

Source	Sum of Squares	DF	Mean Square	F Statistic	P>F
Regr	0.64026384	18	0.035570213	1.15029e+07	0.00000
Error	9.6479564e-07	312	3.0922937e-09		
Total	0.6402648	330			

Appendix L

Spline Function Value X, Y

#filename: Acquire Final Stats Sim.xls

Rank 318 Eqn 6849 Fourier Series Polynomial 9x2

XY	* X Value	Y Value	Y Predict	Residual	Residual%	95% Confidence Limits	95% Prediction Limits	Weights
1	0.0000000	0.0039398	0.0037829	0.0001569	3.9824596	0.0036998 0.0038659	0.0036455 0.0039202	1
2	0.0500000	0.0042072	0.0041888	1.841e-05	0.4374798	0.0041325 0.0042451	0.0040657 0.0043118	1
3	0.1000000	0.0044617	0.0045214	-5.97e-05	-1.337651	0.0044771 0.0045657	0.0044034 0.0046394	1
4	0.1500000	0.0047038	0.0047981	-9.43e-05	-2.004765	0.0047567 0.0048395	0.0046811 0.0049151	1
5	0.2000000	0.0049338	0.0050328	-9.91e-05	-2.007745	0.0049915 0.0050742	0.0049159 0.0051498	1
6	0.2500000	0.0051520	0.0052367	-8.47e-05	-1.644064	0.0051956 0.0052778	0.0051198 0.0053536	1
7	0.3000000	0.0053589	0.0054185	-5.95e-05	-1.111084	0.0053786 0.0054583	0.0053020 0.0055349	1
8	0.3500000	0.0055549	0.0055847	-2.98e-05	-0.536872	0.0055468 0.0056227	0.0054689 0.0057005	1
9	0.4000000	0.0057403	0.0057403	-6.84e-08	-0.001192	0.0057045 0.0057762	0.0056252 0.0058555	1
10	0.4500000	0.0059154	0.0058888	2.661e-05	0.4498996	0.0058549 0.0059228	0.0057743 0.0060034	1
11	0.5000000	0.0060808	0.0060325	4.827e-05	0.7937503	0.0059999 0.0060652	0.0059184 0.0061467	1
12	0.5500000	0.0062367	0.0061729	6.385e-05	1.0237423	0.0061410 0.0062048	0.0060589 0.0062869	1
13	0.6000000	0.0063836	0.0063105	7.303e-05	1.1439677	0.0062789 0.0063422	0.0061966 0.0064245	1
14	0.6500000	0.0065217	0.0064457	7.6e-05	1.1653628	0.0064141 0.0064774	0.0063318 0.0065597	1
15	0.7000000	0.0066516	0.0065783	7.336e-05	1.1028931	0.0065465 0.0066100	0.0064643 0.0066922	1
16	0.7500000	0.0067736	0.0067076	6.594e-05	0.9735093	0.0066759 0.0067393	0.0065937 0.0068215	1
17	0.8000000	0.0068880	0.0068332	5.474e-05	0.7946751	0.0068017 0.0068647	0.0067194 0.0069471	1
18	0.8500000	0.0069952	0.0069544	4.08e-05	0.5833233	0.0069232 0.0069855	0.0068406 0.0070681	1
19	0.9000000	0.0070956	0.0070704	2.52e-05	0.3551366	0.0070398 0.0071010	0.0069568 0.0071841	1
20	0.9500000	0.0071897	0.0071808	8.921e-06	0.1240744	0.0071508 0.0072107	0.0070673 0.0072942	1
21	1.0000000	0.0072777	0.0072848	-7.13e-06	-0.097911	0.0072555 0.0073141	0.0071715 0.0073981	1
22	1.0500000	0.0073601	0.0073822	-2.22e-05	-0.301014	0.0073535 0.0074109	0.0072691 0.0074953	1
23	1.1000000	0.0074372	0.0074727	-3.55e-05	-0.477605	0.0074445 0.0075008	0.0073597 0.0075856	1
24	1.1500000	0.0075094	0.0075561	-4.67e-05	-0.622177	0.0075283 0.0075838	0.0074432 0.0076690	1
25	1.2000000	0.0075771	0.0076325	-5.54e-05	-0.731226	0.0076050 0.0076600	0.0075196 0.0077453	1
26	1.2500000	0.0076406	0.0077020	-6.14e-05	-0.803092	0.0076746 0.0077294	0.0075892 0.0078148	1
27	1.3000000	0.0077004	0.0077649	-6.45e-05	-0.837760	0.0077376 0.0077923	0.0076522 0.0078777	1
28	1.3500000	0.0077569	0.0078218	-6.49e-05	-0.836648	0.0077943 0.0078492	0.0077090 0.0079346	1
29	1.4000000	0.0078103	0.0078730	-6.27e-05	-0.802373	0.0078455 0.0079005	0.0077602 0.0079858	1
30	1.4500000	0.0078612	0.0079192	-5.81e-05	-0.738525	0.0078917 0.0079467	0.0078064 0.0080320	1
31	1.5000000	0.0079098	0.0079611	-5.14e-05	-0.649432	0.0079336 0.0079886	0.0078483 0.0080740	1
32	1.5500000	0.0079565	0.0079995	-4.3e-05	-0.539937	0.0079721 0.0080269	0.0078867 0.0081123	1
33	1.6000000	0.0080018	0.0080350	-3.32e-05	-0.415188	0.0080078 0.0080623	0.0079223 0.0081478	1
34	1.6500000	0.0080460	0.0080685	-2.26e-05	-0.280440	0.0080415 0.0080956	0.0079558 0.0081812	1
35	1.7000000	0.0080894	0.0081008	-1.14e-05	-0.140876	0.0080741 0.0081276	0.0079882 0.0082135	1
36	1.7500000	0.0081326	0.0081327	-1.18e-07	-0.001447	0.0081062 0.0081591	0.0080201 0.0082452	1
37	1.8000000	0.0081757	0.0081648	1.089e-05	0.1332591	0.0081387 0.0081910	0.0080523 0.0082773	1
38	1.8500000	0.0082193	0.0081980	2.13e-05	0.2591345	0.0081722 0.0082239	0.0080856 0.0083104	1
39	1.9000000	0.0082637	0.0082329	3.079e-05	0.3726387	0.0082073 0.0082585	0.0081205 0.0083453	1
40	1.9500000	0.0083093	0.0082701	3.913e-05	0.4708624	0.0082448 0.0082955	0.0081578 0.0083825	1
41	2.0000000	0.0083564	0.0083103	4.609e-05	0.5515685	0.0082851 0.0083355	0.0081980 0.0084226	1
42	2.0500000	0.0084055	0.0083539	5.154e-05	0.6132088	0.0083288 0.0083790	0.0082417 0.0084662	1
43	2.1000000	0.0084569	0.0084015	5.539e-05	0.6549193	0.0083764 0.0084265	0.0082892 0.0085137	1
44	2.1500000	0.0085109	0.0084534	5.758e-05	0.6764953	0.0084283 0.0084784	0.0083411 0.0085656	1
45	2.2000000	0.0085681	0.0085100	5.812e-05	0.6783478	0.0084849 0.0085351	0.0083977 0.0086222	1
46	2.2500000	0.0086287	0.0085716	5.707e-05	0.6614451	0.0085465 0.0085968	0.0084594 0.0086839	1
47	2.3000000	0.0086932	0.0086386	5.453e-05	0.6272402	0.0086134 0.0086639	0.0085264 0.0087509	1
48	2.3500000	0.0087618	0.0087112	5.061e-05	0.5775892	0.0086860 0.0087365	0.0085989 0.0088235	1
49	2.4000000	0.0088351	0.0087896	4.547e-05	0.5146627	0.0087644 0.0088149	0.0086773 0.0089019	1
50	2.4500000	0.0089133	0.0088740	3.929e-05	0.4408533	0.0088488 0.0088993	0.0087618 0.0089863	1
51	2.5000000	0.0089969	0.0089647	3.227e-05	0.3586832	0.0089395 0.0089898	0.0088524 0.0090769	1
52	2.5500000	0.0090862	0.0090616	2.46e-05	0.2707146	0.0090366 0.0090867	0.0089494 0.0091739	1
53	2.6000000	0.0091817	0.0091652	1.648e-05	0.1794660	0.0091403 0.0091901	0.0090530 0.0092774	1
54	2.6500000	0.0092836	0.0092755	8.108e-06	0.0873369	0.0092508 0.0093002	0.0091633 0.0093877	1

55	2.7000000	0.0093924	0.0093927	-3.25e-07	-0.003456	0.0093682	0.0094173	0.0092806	0.0095049	1
56	2.7500000	0.0095084	0.0095171	-8.65e-06	-0.090932	0.0094927	0.0095415	0.0094050	0.0096292	1
57	2.8000000	0.0096321	0.0096488	-1.67e-05	-0.173380	0.0096246	0.0096730	0.0095368	0.0097609	1
58	2.8500000	0.0097638	0.0097881	-2.43e-05	-0.249385	0.0097641	0.0098122	0.0096761	0.0099002	1
59	2.9000000	0.0099039	0.0099353	-3.15e-05	-0.317834	0.0099114	0.0099592	0.0098233	0.0100473	1
60	2.9500000	0.0100527	0.0100907	-3.8e-05	-0.377906	0.0100669	0.0101145	0.0099787	0.0102027	1
61	3.0000000	0.0102107	0.0102545	-4.38e-05	-0.429055	0.0102308	0.0102782	0.0101425	0.0103664	1
62	3.0500000	0.0103782	0.0104271	-4.89e-05	-0.470971	0.0104034	0.0104508	0.0103151	0.0105390	1
63	3.1000000	0.0105556	0.0106088	-5.32e-05	-0.503539	0.0105851	0.0106325	0.0104968	0.0107207	1
64	3.1500000	0.0107433	0.0107999	-5.66e-05	-0.526784	0.0107762	0.0108236	0.0106879	0.0109119	1
65	3.2000000	0.0109417	0.0110008	-5.92e-05	-0.540810	0.0109771	0.0110246	0.0108889	0.0111128	1
66	3.2500000	0.0111511	0.0112119	-6.09e-05	-0.545737	0.0111881	0.0112357	0.0110999	0.0113239	1
67	3.3000000	0.0113719	0.0114335	-6.16e-05	-0.541638	0.0114096	0.0114573	0.0113215	0.0115455	1
68	3.3500000	0.0116045	0.0116659	-6.13e-05	-0.528474	0.0116420	0.0116897	0.0115539	0.0117778	1
69	3.4000000	0.0118493	0.0119093	-6e-05	-0.506035	0.0118854	0.0119332	0.0117973	0.0120213	1
70	3.4500000	0.0121067	0.0121641	-5.74e-05	-0.473887	0.0121402	0.0121880	0.0120521	0.0122761	1
71	3.5000000	0.0123770	0.0124304	-5.34e-05	-0.431325	0.0124065	0.0124543	0.0123184	0.0125424	1
72	3.5500000	0.0126606	0.0127084	-4.78e-05	-0.377338	0.0126846	0.0127323	0.0125964	0.0128204	1
73	3.6000000	0.0129580	0.0129982	-4.02e-05	-0.310579	0.0129745	0.0130220	0.0128863	0.0131102	1
74	3.6500000	0.0132694	0.0132998	-3.04e-05	-0.229353	0.0132761	0.0133235	0.0131879	0.0134118	1
75	3.7000000	0.0135952	0.0136131	-1.8e-05	-0.132217	0.0135896	0.0136367	0.0135012	0.0137251	1
76	3.7500000	0.0139347	0.0139380	-3.36e-06	-0.024139	0.0139146	0.0139615	0.0138261	0.0140499	1
77	3.8000000	0.0142866	0.0142742	1.239e-05	0.0867202	0.0142509	0.0142976	0.0141624	0.0143861	1
78	3.8500000	0.0146498	0.0146214	2.837e-05	0.1936310	0.0145982	0.0146446	0.0145096	0.0147333	1
79	3.9000000	0.0150228	0.0149790	4.376e-05	0.2912853	0.0149559	0.0150022	0.0148672	0.0150909	1
80	3.9500000	0.0154045	0.0153466	5.787e-05	0.3756864	0.0153236	0.0153696	0.0152348	0.0154584	1
81	4.0000000	0.0157935	0.0157234	7.013e-05	0.4440280	0.0157004	0.0157463	0.0156116	0.0158352	1
82	4.0500000	0.0161886	0.0161085	8.006e-05	0.4945664	0.0160856	0.0161314	0.0159967	0.0162203	1
83	4.1000000	0.0165885	0.0165011	8.734e-05	0.5264908	0.0164783	0.0165240	0.0163894	0.0166129	1
84	4.1500000	0.0169919	0.0169001	9.172e-05	0.5397966	0.0168773	0.0169230	0.0167884	0.0170119	1
85	4.2000000	0.0173975	0.0173044	9.31e-05	0.5351624	0.0172815	0.0173273	0.0171926	0.0174162	1
86	4.2500000	0.0178041	0.0177126	9.148e-05	0.5138336	0.0176897	0.0177355	0.0176009	0.0178244	1
87	4.3000000	0.0182104	0.0181235	8.696e-05	0.4775135	0.0181005	0.0181464	0.0180117	0.0182353	1
88	4.3500000	0.0186151	0.0185354	7.972e-05	0.4282616	0.0185124	0.0185584	0.0184236	0.0186472	1
89	4.4000000	0.0190170	0.0189469	7.006e-05	0.3684003	0.0189239	0.0189699	0.0188351	0.0190587	1
90	4.4500000	0.0194147	0.0193563	5.833e-05	0.3004281	0.0193333	0.0193794	0.0192445	0.0194682	1
91	4.5000000	0.0198070	0.0197620	4.495e-05	0.2269412	0.0197389	0.0197851	0.0196502	0.0198738	1
92	4.5500000	0.0201925	0.0201621	3.04e-05	0.1505611	0.0201390	0.0201852	0.0200503	0.0202740	1
93	4.6000000	0.0205702	0.0205550	1.519e-05	0.0738683	0.0205319	0.0205781	0.0204431	0.0206668	1
94	4.6500000	0.0209385	0.0209387	-1.38e-07	-0.000658	0.0209156	0.0209617	0.0208268	0.0210505	1
95	4.7000000	0.0212964	0.0213114	-1.51e-05	-0.070697	0.0212884	0.0213345	0.0211996	0.0214232	1
96	4.7500000	0.0216424	0.0216714	-2.9e-05	-0.134134	0.0216485	0.0216944	0.0215596	0.0217832	1
97	4.8000000	0.0219754	0.0220169	-4.16e-05	-0.189114	0.0219940	0.0220398	0.0219051	0.0221287	1
98	4.8500000	0.0222940	0.0223462	-5.22e-05	-0.234080	0.0223233	0.0223690	0.0222344	0.0224579	1
99	4.9000000	0.0225969	0.0226575	-6.05e-05	-0.267819	0.0226347	0.0226802	0.0225457	0.0227692	1
100	4.9500000	0.0228830	0.0229493	-6.62e-05	-0.289501	0.0229266	0.0229719	0.0228375	0.0230610	1
101	5.0000000	0.0231509	0.0232200	-6.92e-05	-0.298719	0.0231975	0.0232426	0.0231083	0.0233318	1
102	5.0500000	0.0233993	0.0234685	-6.92e-05	-0.295525	0.0234460	0.0234909	0.0233567	0.0235802	1
103	5.1000000	0.0236270	0.0236932	-6.63e-05	-0.280469	0.0236708	0.0237157	0.0235816	0.0238049	1
104	5.1500000	0.0238326	0.0238933	-6.07e-05	-0.254639	0.0238709	0.0239157	0.0237816	0.0240050	1
105	5.2000000	0.0240150	0.0240678	-5.28e-05	-0.219701	0.0240454	0.0240901	0.0239561	0.0241794	1
106	5.2500000	0.0241728	0.0242158	-4.3e-05	-0.177943	0.0241935	0.0242381	0.0241041	0.0243275	1
107	5.3000000	0.0243047	0.0243369	-3.22e-05	-0.132321	0.0243146	0.0243592	0.0242252	0.0244486	1
108	5.3500000	0.0244096	0.0244307	-2.11e-05	-0.086512	0.0244083	0.0244530	0.0243190	0.0245424	1
109	5.4000000	0.0244860	0.0244970	-1.1e-05	-0.044974	0.0244746	0.0245194	0.0243853	0.0246087	1
110	5.4500000	0.0245327	0.0245359	-3.19e-06	-0.013011	0.0245135	0.0245583	0.0244242	0.0246476	1
111	5.5000000	0.0245485	0.0245478	7.722e-07	0.0031455	0.0245253	0.0245702	0.0244361	0.0246595	1
112	5.5500000	0.0245328	0.0245330	-2.76e-07	-0.001123	0.0245105	0.0245555	0.0244213	0.0246447	1
113	5.6000000	0.0244873	0.0244925	-5.14e-06	-0.020983	0.0244699	0.0245150	0.0243808	0.0246042	1
114	5.6500000	0.0244149	0.0244271	-1.22e-05	-0.049989	0.0244045	0.0244497	0.0243154	0.0245388	1
115	5.7000000	0.0243180	0.0243381	-2.01e-05	-0.082641	0.0243155	0.0243607	0.0242264	0.0244498	1
116	5.7500000	0.0241993	0.0242270	-2.77e-05	-0.114311	0.0242044	0.0242496	0.0241152	0.0243387	1
117	5.8000000	0.0240614	0.0240953	-3.4e-05	-0.141196	0.0240727	0.0241179	0.0239836	0.0242071	1
118	5.8500000	0.0239068	0.0239451	-3.83e-05	-0.160277	0.0239225	0.0239677	0.0238334	0.0240568	1
119	5.9000000	0.0237382	0.0237784	-4.02e-05	-0.169283	0.0237559	0.0238010	0.0236667	0.0238901	1
120	5.9500000	0.0235582	0.0235975	-3.93e-05	-0.166669	0.0235750	0.0236200	0.0234858	0.0237092	1
121	6.0000000	0.0233694	0.0234048	-3.54e-05	-0.151582	0.0233823	0.0234272	0.0232931	0.0235165	1

122	6.0500000	0.0231743	0.0232030	-2.87e-05	-0.123844	0.0231806	0.0232254	0.0230913	0.0233147	1
123	6.1000000	0.0229756	0.0229949	-1.93e-05	-0.083912	0.0229726	0.0230172	0.0228832	0.0231065	1
124	6.1500000	0.0227759	0.0227834	-7.48e-06	-0.032853	0.0227611	0.0228056	0.0226717	0.0228950	1
125	6.2000000	0.0225777	0.0225715	6.255e-06	0.0277058	0.0225493	0.0225937	0.0224598	0.0226831	1
126	6.2500000	0.0223838	0.0223624	2.14e-05	0.0956269	0.0223402	0.0223845	0.0222507	0.0224740	1
127	6.3000000	0.0221966	0.0221592	3.736e-05	0.1683261	0.0221371	0.0221813	0.0220476	0.0222708	1
128	6.3500000	0.0220188	0.0219653	5.347e-05	0.2428520	0.0219432	0.0219873	0.0218537	0.0220769	1
129	6.4000000	0.0218529	0.0217839	6.905e-05	0.3159789	0.0217619	0.0218059	0.0216723	0.0218955	1
130	6.4500000	0.0217017	0.0216183	8.34e-05	0.3843154	0.0215962	0.0216403	0.0215066	0.0217299	1
131	6.5000000	0.0215676	0.0214717	9.585e-05	0.4444283	0.0214497	0.0214937	0.0213601	0.0215833	1
132	6.5500000	0.0214533	0.0213475	0.0001058	0.4929814	0.0213255	0.0213695	0.0212359	0.0214591	1
133	6.6000000	0.0213613	0.0212488	0.0001125	0.5268842	0.0212267	0.0212709	0.0211372	0.0213604	1
134	6.6500000	0.0212944	0.0211787	0.0001157	0.5434502	0.0211565	0.0212008	0.0210670	0.0212903	1
135	6.7000000	0.0212550	0.0211401	0.0001149	0.5405560	0.0211180	0.0211623	0.0210285	0.0212518	1
136	6.7500000	0.0212458	0.0211360	0.0001098	0.5167970	0.0211138	0.0211582	0.0210244	0.0212477	1
137	6.8000000	0.0212694	0.0211691	0.0001003	0.4716310	0.0211468	0.0211913	0.0210574	0.0212807	1
138	6.8500000	0.0213284	0.0212419	8.649e-05	0.4055018	0.0212196	0.0212642	0.0211302	0.0213535	1
139	6.9000000	0.0214253	0.0213568	6.855e-05	0.3199333	0.0213345	0.0213791	0.0212451	0.0214684	1
140	6.9500000	0.0215628	0.0215159	4.692e-05	0.2175880	0.0214936	0.0215383	0.0214043	0.0216276	1
141	7.0000000	0.0217436	0.0217213	2.224e-05	0.1022812	0.0216990	0.0217437	0.0216096	0.0218330	1
142	7.0500000	0.0219701	0.0219747	-4.62e-06	-0.021051	0.0219523	0.0219970	0.0218630	0.0220863	1
143	7.1000000	0.0222449	0.0222775	-3.26e-05	-0.146435	0.0222552	0.0222998	0.0221658	0.0223892	1
144	7.1500000	0.0225708	0.0226310	-6.03e-05	-0.266988	0.0226088	0.0226533	0.0225194	0.0227427	1
145	7.2000000	0.0229502	0.0230363	-8.61e-05	-0.375091	0.0230141	0.0230585	0.0229247	0.0231480	1
146	7.2500000	0.0233858	0.0234940	-0.000108	-0.462598	0.0234718	0.0235162	0.0233824	0.0236057	1
147	7.3000000	0.0238803	0.0240047	-0.000124	-0.521066	0.0239826	0.0240268	0.0238931	0.0241163	1
148	7.3500000	0.0244360	0.0245685	-0.000132	-0.542132	0.0245464	0.0245906	0.0244569	0.0246801	1
149	7.4000000	0.0250538	0.0251854	-0.000132	-0.525248	0.0251634	0.0252074	0.0250738	0.0252970	1
150	7.4500000	0.0257313	0.0258551	-0.000124	-0.480961	0.0258331	0.0258770	0.0257435	0.0259667	1
151	7.5000000	0.0264660	0.0265769	-0.000111	-0.419136	0.0265550	0.0265989	0.0264654	0.0266885	1
152	7.5500000	0.0272554	0.0273502	-9.48e-05	-0.347894	0.0273283	0.0273721	0.0272386	0.0274618	1
153	7.6000000	0.0280969	0.0281738	-7.69e-05	-0.273690	0.0281519	0.0281956	0.0280622	0.0282854	1
154	7.6500000	0.0289879	0.0290463	-5.84e-05	-0.201454	0.0290245	0.0290682	0.0289348	0.0291579	1
155	7.7000000	0.0299261	0.0299664	-4.03e-05	-0.134747	0.0299445	0.0299883	0.0298548	0.0300780	1
156	7.7500000	0.0309087	0.0309322	-2.35e-05	-0.075957	0.0309103	0.0309541	0.0308206	0.0310438	1
157	7.8000000	0.0319333	0.0319418	-8.45e-06	-0.026475	0.0319198	0.0319637	0.0318302	0.0320534	1
158	7.8500000	0.0329974	0.0329930	4.329e-06	0.0131180	0.0329711	0.0330150	0.0328814	0.0331046	1
159	7.9000000	0.0340983	0.0340837	1.462e-05	0.0428835	0.0340616	0.0341057	0.0339721	0.0341953	1
160	7.9500000	0.0352336	0.0352113	2.233e-05	0.0633742	0.0351892	0.0352333	0.0350996	0.0353229	1
161	8.0000000	0.0364007	0.0363732	2.748e-05	0.0754993	0.0363511	0.0363953	0.0362616	0.0364849	1
162	8.0500000	0.0375971	0.0375669	3.023e-05	0.0804085	0.0375447	0.0375891	0.0374552	0.0376785	1
163	8.1000000	0.0388203	0.0387894	3.082e-05	0.0793947	0.0387672	0.0388117	0.0386778	0.0389011	1
164	8.1500000	0.0400676	0.0400381	2.957e-05	0.0738126	0.0400158	0.0400603	0.0399264	0.0401497	1
165	8.2000000	0.0413367	0.0413098	2.687e-05	0.0650132	0.0412875	0.0413321	0.0411981	0.0414215	1
166	8.2500000	0.0426248	0.0426017	2.314e-05	0.0542906	0.0425794	0.0426240	0.0424900	0.0427134	1
167	8.3000000	0.0439296	0.0439108	1.882e-05	0.0428416	0.0438885	0.0439331	0.0437991	0.0440225	1
168	8.3500000	0.0452484	0.0452341	1.436e-05	0.0317344	0.0452118	0.0452563	0.0451224	0.0453457	1
169	8.4000000	0.0465788	0.0465686	1.019e-05	0.0218855	0.0465464	0.0465908	0.0464570	0.0466803	1
170	8.4500000	0.0479181	0.0479114	6.73e-06	0.0140446	0.0478892	0.0479336	0.0477998	0.0480230	1
171	8.5000000	0.0492639	0.0492596	4.327e-06	0.0087837	0.0492375	0.0492817	0.0491480	0.0493712	1
172	8.5500000	0.0506136	0.0506103	3.286e-06	0.0064922	0.0505883	0.0506324	0.0504987	0.0507220	1
173	8.6000000	0.0519647	0.0519609	3.832e-06	0.0073746	0.0519388	0.0519829	0.0518493	0.0520725	1
174	8.6500000	0.0533146	0.0533085	6.106e-06	0.0114519	0.0532865	0.0533305	0.0531969	0.0534201	1
175	8.7000000	0.0546608	0.0546507	1.015e-05	0.0185648	0.0546287	0.0546726	0.0545391	0.0547623	1
176	8.7500000	0.0560008	0.0559849	1.589e-05	0.0283790	0.0559630	0.0560068	0.0558733	0.0560965	1
177	8.8000000	0.0573320	0.0573088	2.316e-05	0.0403909	0.0572869	0.0573307	0.0571972	0.0574204	1
178	8.8500000	0.0586518	0.0586202	3.163e-05	0.0539352	0.0585983	0.0586421	0.0585086	0.0587318	1
179	8.9000000	0.0599579	0.0599170	4.089e-05	0.0681923	0.0598951	0.0599389	0.0598054	0.0600286	1
180	8.9500000	0.0612475	0.0611972	5.034e-05	0.0821960	0.0611753	0.0612191	0.0610856	0.0613087	1
181	9.0000000	0.0625182	0.0624589	5.929e-05	0.0948414	0.0624370	0.0624808	0.0623473	0.0625705	1
182	9.0500000	0.0637675	0.0637006	6.689e-05	0.1048925	0.0636786	0.0637225	0.0635890	0.0638122	1
183	9.1000000	0.0649927	0.0649206	7.214e-05	0.1109894	0.0648985	0.0649426	0.0648090	0.0650322	1
184	9.1500000	0.0661914	0.0661175	7.391e-05	0.1116549	0.0660954	0.0661396	0.0660059	0.0662291	1
185	9.2000000	0.0673612	0.0672901	7.106e-05	0.1054935	0.0672680	0.0673122	0.0671785	0.0674017	1
186	9.2500000	0.0685011	0.0684372	6.384e-05	0.0931982	0.0684150	0.0684594	0.0683256	0.0685489	1
187	9.3000000	0.0696112	0.0695579	5.333e-05	0.0766124	0.0695357	0.0695802	0.0694463	0.0696696	1
188	9.3500000	0.0706917	0.0706512	4.05e-05	0.0572950	0.0706290	0.0706735	0.0705396	0.0707629	1

189	9.4000000	0.0717427	0.0717165	2.621e-05	0.0365345	0.0716942	0.0717388	0.0716048	0.0718281	1
190	9.4500000	0.0727642	0.0727530	1.119e-05	0.0153821	0.0727306	0.0727753	0.0726413	0.0728647	1
191	9.5000000	0.0737563	0.0737602	-3.92e-06	-0.005317	0.0737379	0.0737826	0.0736485	0.0738719	1
192	9.5500000	0.0747192	0.0747378	-1.86e-05	-0.024897	0.0747154	0.0747601	0.0746261	0.0748494	1
193	9.6000000	0.0756528	0.0756853	-3.24e-05	-0.042848	0.0756629	0.0757076	0.0755736	0.0757969	1
194	9.6500000	0.0765575	0.0766025	-4.5e-05	-0.058791	0.0765802	0.0766248	0.0764908	0.0767141	1
195	9.7000000	0.0774331	0.0774892	-5.61e-05	-0.072462	0.0774670	0.0775115	0.0773776	0.0776009	1
196	9.7500000	0.0782799	0.0783454	-6.55e-05	-0.083693	0.0783232	0.0783677	0.0782338	0.0784571	1
197	9.8000000	0.0790980	0.0791710	-7.31e-05	-0.092396	0.0791489	0.0791932	0.0790594	0.0792827	1
198	9.8500000	0.0798873	0.0799660	-7.87e-05	-0.098549	0.0799439	0.0799882	0.0798544	0.0800777	1
199	9.9000000	0.0806481	0.0807305	-8.24e-05	-0.102185	0.0807084	0.0807526	0.0806189	0.0808421	1
200	9.9500000	0.0813804	0.0814645	-8.41e-05	-0.103380	0.0814425	0.0814866	0.0813529	0.0815762	1
201	10.000000	0.0820844	0.0821683	-8.39e-05	-0.102244	0.0821463	0.0821903	0.0820567	0.0822799	1
202	10.050000	0.0827600	0.0828419	-8.19e-05	-0.098917	0.0828199	0.0828639	0.0827303	0.0829535	1
203	10.100000	0.0834075	0.0834855	-7.8e-05	-0.093556	0.0834635	0.0835076	0.0833739	0.0835972	1
204	10.150000	0.0840269	0.0840995	-7.25e-05	-0.086336	0.0840774	0.0841215	0.0839879	0.0842111	1
205	10.200000	0.0846184	0.0846839	-6.55e-05	-0.077443	0.0846618	0.0847060	0.0845723	0.0847955	1
206	10.250000	0.0851819	0.0852390	-5.71e-05	-0.067073	0.0852169	0.0852612	0.0851274	0.0853507	1
207	10.300000	0.0857177	0.0857652	-4.75e-05	-0.055429	0.0857430	0.0857874	0.0856536	0.0858768	1
208	10.350000	0.0862258	0.0862626	-3.68e-05	-0.042724	0.0862404	0.0862849	0.0861510	0.0863743	1
209	10.400000	0.0867063	0.0867316	-2.53e-05	-0.029178	0.0867093	0.0867539	0.0866199	0.0868433	1
210	10.450000	0.0871593	0.0871724	-1.31e-05	-0.015020	0.0871500	0.0871948	0.0870607	0.0872841	1
211	10.500000	0.0875850	0.0875854	-4.32e-07	-0.000494	0.0875630	0.0876079	0.0874737	0.0876971	1
212	10.550000	0.0879833	0.0879709	1.244e-05	0.0141425	0.0879484	0.0879934	0.0878592	0.0880826	1
213	10.600000	0.0883545	0.0883292	2.528e-05	0.0286099	0.0883067	0.0883518	0.0882175	0.0884409	1
214	10.650000	0.0886986	0.0886608	3.779e-05	0.0426048	0.0886382	0.0886834	0.0885491	0.0887725	1
215	10.700000	0.0890157	0.0889660	4.967e-05	0.0557936	0.0889434	0.0889886	0.0888543	0.0890777	1
216	10.750000	0.0893059	0.0892453	6.056e-05	0.0678069	0.0892227	0.0892679	0.0891336	0.0893571	1
217	10.800000	0.0895693	0.0894992	7.007e-05	0.0782328	0.0894766	0.0895218	0.0893875	0.0896109	1
218	10.850000	0.0898060	0.0897282	7.778e-05	0.0866112	0.0897057	0.0897508	0.0896165	0.0898399	1
219	10.900000	0.0900161	0.0899329	8.32e-05	0.0924263	0.0899104	0.0899555	0.0898212	0.0900446	1
220	10.950000	0.0901998	0.0901140	8.578e-05	0.0950999	0.0900915	0.0901365	0.0900023	0.0902257	1
221	11.000000	0.0903570	0.0902721	8.492e-05	0.0939845	0.0902496	0.0902945	0.0901604	0.0903838	1
222	11.050000	0.0904882	0.0904080	8.023e-05	0.0886685	0.0903856	0.0904304	0.0902963	0.0905197	1
223	11.100000	0.0905949	0.0905225	7.239e-05	0.0799042	0.0905002	0.0905449	0.0904109	0.0906342	1
224	11.150000	0.0906790	0.0906167	6.226e-05	0.0686619	0.0905943	0.0906390	0.0905050	0.0907284	1
225	11.200000	0.0907421	0.0906914	5.065e-05	0.0558153	0.0906691	0.0907137	0.0905797	0.0908031	1
226	11.250000	0.0907861	0.0907478	3.826e-05	0.0421412	0.0907255	0.0907701	0.0906361	0.0908595	1
227	11.300000	0.0908127	0.0907870	2.572e-05	0.0283188	0.0907647	0.0908094	0.0906753	0.0908987	1
228	11.350000	0.0908239	0.0908103	1.356e-05	0.0149293	0.0907880	0.0908327	0.0906987	0.0909220	1
229	11.400000	0.0908213	0.0908191	2.231e-06	0.0024563	0.0907967	0.0908415	0.0907074	0.0909308	1
230	11.450000	0.0908069	0.0908148	-7.91e-06	-0.008714	0.0907923	0.0908373	0.0907031	0.0909265	1
231	11.500000	0.0907823	0.0907989	-1.66e-05	-0.018289	0.0907763	0.0908214	0.0906871	0.0909106	1
232	11.550000	0.0907493	0.0907730	-2.37e-05	-0.026072	0.0907503	0.0907956	0.0906612	0.0908847	1
233	11.600000	0.0907098	0.0907388	-2.9e-05	-0.031955	0.0907161	0.0907616	0.0906271	0.0908506	1
234	11.650000	0.0906656	0.0906982	-3.26e-05	-0.035913	0.0906754	0.0907210	0.0905864	0.0908100	1
235	11.700000	0.0906185	0.0906529	-3.44e-05	-0.038003	0.0906300	0.0906758	0.0905411	0.0907647	1
236	11.750000	0.0905702	0.0906049	-3.47e-05	-0.038355	0.0905819	0.0906279	0.0904931	0.0907167	1
237	11.800000	0.0905225	0.0905562	-3.36e-05	-0.037164	0.0905332	0.0905792	0.0904444	0.0906680	1
238	11.850000	0.0904773	0.0905087	-3.14e-05	-0.034680	0.0904857	0.0905318	0.0903969	0.0906205	1
239	11.900000	0.0904364	0.0904646	-2.82e-05	-0.031200	0.0904415	0.0904877	0.0903528	0.0905764	1
240	11.950000	0.0904015	0.0904259	-2.45e-05	-0.027055	0.0904029	0.0904490	0.0903141	0.0905378	1
241	12.000000	0.0903744	0.0903948	-2.04e-05	-0.022596	0.0903718	0.0904179	0.0902830	0.0905067	1
242	12.050000	0.0903570	0.0903734	-1.64e-05	-0.018185	0.0903504	0.0903965	0.0902616	0.0904852	1
243	12.100000	0.0903510	0.0903638	-1.28e-05	-0.014174	0.0903408	0.0903868	0.0902520	0.0904756	1
244	12.150000	0.0903582	0.0903681	-9.85e-06	-0.010897	0.0903451	0.0903910	0.0902563	0.0904799	1
245	12.200000	0.0903805	0.0903883	-7.82e-06	-0.008649	0.0903654	0.0904112	0.0902765	0.0905001	1
246	12.250000	0.0904196	0.0904265	-6.94e-06	-0.007672	0.0904036	0.0904494	0.0903147	0.0905383	1
247	12.300000	0.0904773	0.0904846	-7.36e-06	-0.008138	0.0904618	0.0905075	0.0903728	0.0905964	1
248	12.350000	0.0905554	0.0905645	-9.18e-06	-0.010137	0.0905417	0.0905874	0.0904528	0.0906763	1
249	12.400000	0.0906557	0.0906681	-1.24e-05	-0.013656	0.0906452	0.0906909	0.0905563	0.0907798	1
250	12.450000	0.0907800	0.0907969	-1.69e-05	-0.018569	0.0907740	0.0908198	0.0906851	0.0909087	1
251	12.500000	0.0909302	0.0909525	-2.24e-05	-0.024622	0.0909296	0.0909755	0.0908408	0.0910643	1
252	12.550000	0.0911079	0.0911365	-2.86e-05	-0.031421	0.0911135	0.0911595	0.0910247	0.0912483	1
253	12.600000	0.0913150	0.0913501	-3.51e-05	-0.038420	0.0913270	0.0913732	0.0912383	0.0914620	1
254	12.650000	0.0915534	0.0915945	-4.11e-05	-0.044917	0.0915713	0.0916177	0.0914827	0.0917063	1
255	12.700000	0.0918247	0.0918707	-4.6e-05	-0.050045	0.0918473	0.0918940	0.0917588	0.0919825	1

256	12.750000	0.0921308	0.0921794	-4.86e-05	-0.052769	0.0921560	0.0922029	0.0920675	0.0922913	1
257	12.800000	0.0924735	0.0925215	-4.8e-05	-0.051887	0.0924979	0.0925451	0.0924096	0.0926334	1
258	12.850000	0.0928546	0.0928973	-4.28e-05	-0.046050	0.0928737	0.0929210	0.0927854	0.0930093	1
259	12.900000	0.0932746	0.0933073	-3.27e-05	-0.035048	0.0932835	0.0933310	0.0931953	0.0934192	1
260	12.950000	0.0937322	0.0937514	-1.92e-05	-0.020493	0.0937276	0.0937752	0.0936394	0.0938634	1
261	13.000000	0.0942259	0.0942297	-3.72e-06	-0.003951	0.0942058	0.0942535	0.0941177	0.0943417	1
262	13.050000	0.0947544	0.0947418	1.254e-05	0.0132351	0.0947179	0.0947657	0.0946298	0.0948538	1
263	13.100000	0.0953160	0.0952875	2.854e-05	0.0299398	0.0952635	0.0953114	0.0951755	0.0953995	1
264	13.150000	0.0959094	0.0958660	4.34e-05	0.0452509	0.0958421	0.0958898	0.0957540	0.0959779	1
265	13.200000	0.0965330	0.0964765	5.643e-05	0.0584591	0.0964527	0.0965004	0.0963645	0.0965885	1
266	13.250000	0.0971854	0.0971183	6.71e-05	0.0690469	0.0970945	0.0971421	0.0970063	0.0972302	1
267	13.300000	0.0978651	0.0977900	7.504e-05	0.0766746	0.0977663	0.0978138	0.0976781	0.0979020	1
268	13.350000	0.0985706	0.0984906	8.001e-05	0.0811653	0.0984669	0.0985143	0.0983787	0.0986026	1
269	13.400000	0.0993005	0.0992186	8.191e-05	0.0824876	0.0991949	0.0992423	0.0991067	0.0993306	1
270	13.450000	0.1000534	0.0999726	8.078e-05	0.0807381	0.0999489	0.0999963	0.0998606	0.1000845	1
271	13.500000	0.1008276	0.1007508	7.675e-05	0.0761228	0.1007271	0.1007746	0.1006389	0.1008628	1
272	13.550000	0.1016218	0.1015517	7.006e-05	0.0689386	0.1015279	0.1015755	0.1014398	0.1016637	1
273	13.600000	0.1024345	0.1023735	6.1e-05	0.0595540	0.1023496	0.1023974	0.1022615	0.1024854	1
274	13.650000	0.1032641	0.1032142	4.997e-05	0.0483913	0.1031901	0.1032382	0.1031021	0.1033262	1
275	13.700000	0.1041094	0.1040720	3.738e-05	0.0359087	0.1040478	0.1040962	0.1039599	0.1041840	1
276	13.750000	0.1049686	0.1049449	2.371e-05	0.0225832	0.1049206	0.1049693	0.1048328	0.1050570	1
277	13.800000	0.1058405	0.1058311	9.414e-06	0.0088946	0.1058065	0.1058556	0.1057190	0.1059432	1
278	13.850000	0.1067235	0.1067285	-5e-06	-0.004689	0.1067038	0.1067532	0.1066163	0.1068407	1
279	13.900000	0.1076161	0.1076352	-1.91e-05	-0.017723	0.1076103	0.1076601	0.1075230	0.1077474	1
280	13.950000	0.1085169	0.1085493	-3.23e-05	-0.029800	0.1085242	0.1085743	0.1084370	0.1086615	1
281	14.000000	0.1094245	0.1094688	-4.44e-05	-0.040560	0.1094437	0.1094940	0.1093566	0.1095811	1
282	14.050000	0.1103372	0.1103920	-5.48e-05	-0.049696	0.1103668	0.1104173	0.1102798	0.1105043	1
283	14.100000	0.1112537	0.1113171	-6.34e-05	-0.056961	0.1112918	0.1113423	0.1112048	0.1114294	1
284	14.150000	0.1121725	0.1122423	-6.97e-05	-0.062175	0.1122170	0.1122675	0.1121300	0.1123545	1
285	14.200000	0.1130921	0.1131659	-7.38e-05	-0.065226	0.1131407	0.1131911	0.1130536	0.1132782	1
286	14.250000	0.1140111	0.1140864	-7.53e-05	-0.066071	0.1140612	0.1141116	0.1139741	0.1141987	1
287	14.300000	0.1149279	0.1150023	-7.44e-05	-0.064736	0.1149772	0.1150274	0.1148900	0.1151146	1
288	14.350000	0.1158411	0.1159121	-7.1e-05	-0.061311	0.1158871	0.1159372	0.1157999	0.1160244	1
289	14.400000	0.1167493	0.1168146	-6.53e-05	-0.055952	0.1167895	0.1168397	0.1167023	0.1169268	1
290	14.450000	0.1176509	0.1177084	-5.75e-05	-0.048868	0.1176833	0.1177335	0.1175961	0.1178206	1
291	14.500000	0.1185445	0.1185923	-4.78e-05	-0.040322	0.1185671	0.1186175	0.1184800	0.1187045	1
292	14.550000	0.1194286	0.1194651	-3.66e-05	-0.030616	0.1194398	0.1194905	0.1193528	0.1195774	1
293	14.600000	0.1203017	0.1203259	-2.42e-05	-0.020087	0.1203003	0.1203514	0.1202135	0.1204382	1
294	14.650000	0.1211624	0.1211734	-1.1e-05	-0.009098	0.1211476	0.1211992	0.1210610	0.1212858	1
295	14.700000	0.1220092	0.1220068	2.414e-06	0.0019789	0.1219806	0.1220329	0.1218943	0.1221193	1
296	14.750000	0.1228406	0.1228249	1.568e-05	0.0127649	0.1227985	0.1228514	0.1227124	0.1229375	1
297	14.800000	0.1236552	0.1236269	2.831e-05	0.0228914	0.1236001	0.1236536	0.1235142	0.1237395	1
298	14.850000	0.1244514	0.1244116	3.983e-05	0.0320084	0.1243846	0.1244386	0.1242989	0.1245243	1
299	14.900000	0.1252279	0.1251780	4.984e-05	0.0397963	0.1251508	0.1252053	0.1250653	0.1252908	1
300	14.950000	0.1259831	0.1259251	5.792e-05	0.0459758	0.1258977	0.1259526	0.1258123	0.1260379	1
301	15.000000	0.1267155	0.1266517	6.376e-05	0.0503177	0.1266242	0.1266792	0.1265389	0.1267646	1
302	15.050000	0.1274237	0.1273567	6.709e-05	0.0526520	0.1273291	0.1273842	0.1272438	0.1274695	1
303	15.100000	0.1281063	0.1280386	6.774e-05	0.0528754	0.1280111	0.1280661	0.1279257	0.1281514	1
304	15.150000	0.1287617	0.1286961	6.561e-05	0.0509573	0.1286687	0.1287235	0.1285833	0.1288089	1
305	15.200000	0.1293885	0.1293277	6.074e-05	0.0469448	0.1293003	0.1293551	0.1292149	0.1294405	1
306	15.250000	0.1299851	0.1299319	5.325e-05	0.0409653	0.1299045	0.1299593	0.1298191	0.1300447	1
307	15.300000	0.1305503	0.1305069	4.338e-05	0.0332270	0.1304794	0.1305344	0.1303941	0.1306197	1
308	15.350000	0.1310823	0.1310508	3.148e-05	0.0240177	0.1310231	0.1310786	0.1309380	0.1311637	1
309	15.400000	0.1315799	0.1315618	1.803e-05	0.0136995	0.1315337	0.1315900	0.1314489	0.1316748	1
310	15.450000	0.1320414	0.1320379	3.569e-06	0.0027029	0.1320092	0.1320665	0.1319247	0.1321510	1
311	15.500000	0.1324655	0.1324768	-1.12e-05	-0.008484	0.1324475	0.1325061	0.1323635	0.1325900	1
312	15.550000	0.1328507	0.1328764	-2.57e-05	-0.019329	0.1328464	0.1329063	0.1327629	0.1329898	1
313	15.600000	0.1331955	0.1332344	-3.9e-05	-0.029273	0.1332038	0.1332651	0.1331208	0.1333481	1
314	15.650000	0.1334983	0.1335487	-5.04e-05	-0.037747	0.1335176	0.1335799	0.1334350	0.1336625	1
315	15.700000	0.1337579	0.1338170	-5.91e-05	-0.044202	0.1337855	0.1338485	0.1337031	0.1339309	1
316	15.750000	0.1339726	0.1340371	-6.45e-05	-0.048132	0.1340054	0.1340688	0.1339232	0.1341510	1
317	15.800000	0.1341410	0.1342069	-6.59e-05	-0.049109	0.1341752	0.1342387	0.1340930	0.1343208	1
318	15.850000	0.1342617	0.1343246	-6.29e-05	-0.046815	0.1342929	0.1343562	0.1342107	0.1344385	1
319	15.900000	0.1343331	0.1343883	-5.52e-05	-0.041092	0.1343567	0.1344200	0.1342744	0.1345023	1
320	15.950000	0.1343539	0.1343968	-4.3e-05	-0.031974	0.1343649	0.1344288	0.1342829	0.1345108	1
321	16.000000	0.1343224	0.1343489	-2.65e-05	-0.019749	0.1343163	0.1343816	0.1342348	0.1344631	1
322	16.050000	0.1342373	0.1342440	-6.72e-06	-0.005005	0.1342101	0.1342780	0.1341295	0.1343586	1

323	16.100000	0.1340971	0.1340819	1.517e-05	0.0113102	0.1340461	0.1341177	0.1339668	0.1341970	1
324	16.150000	0.1339003	0.1338630	3.725e-05	0.0278208	0.1338251	0.1339010	0.1337472	0.1339788	1
325	16.200000	0.1336454	0.1335883	5.701e-05	0.0426604	0.1335485	0.1336282	0.1334719	0.1337048	1
326	16.250000	0.1333309	0.1332597	7.119e-05	0.0533964	0.1332186	0.1333008	0.1331429	0.1333766	1
327	16.300000	0.1329555	0.1328798	7.572e-05	0.0569530	0.1328384	0.1329211	0.1327628	0.1329967	1
328	16.350000	0.1325175	0.1324519	6.563e-05	0.0495287	0.1324105	0.1324933	0.1323349	0.1325689	1
329	16.400000	0.1320157	0.1319807	3.5e-05	0.0265091	0.1319364	0.1320249	0.1318626	0.1320987	1
330	16.450000	0.1314483	0.1314715	-2.32e-05	-0.017625	0.1314152	0.1315278	0.1313484	0.1315946	1
331	16.500000	0.1308141	0.1309311	-0.000117	-0.089393	0.1308480	0.1310141	0.1307937	0.1310684	1