UNIVERSITÀ DEGLI STUDI DI CAGLIARI
FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA MECCANICA
XV CICLO DOTTORATO IN PROGETTAZIONE MECCANICA

HAPTIC AND VISUAL SIMULATION OF BONE DISSECTION

Relatori
Prof. Bruno Picasso
Dr. GianLuigi Zanetti
Dr. Enrico Gobbetti

Tesi di Dottorato di
Marco Agus

Marzo 2004

In bone dissection virtual simulation, force restitution represents the key to realistically mimicking a patient–specific operating environment. The force is rendered using haptic devices controlled by parametrized mathematical models that represent the bone–burr contact. This dissertation presents and discusses a haptic simulation of a bone cutting burr, that it is being developed as a component of a training system for temporal bone surgery. A physically based model was used to describe the burr–bone interaction, including haptic forces evaluation, bone erosion process and resulting debris. The model was experimentally validated and calibrated by employing a custom experimental set–up consisting of a force–controlled robot arm holding a high–speed rotating tool and a contact force measuring apparatus. Psychophysical testing was also carried out to assess individual reaction to the haptic environment. The results suggest that the simulator is capable of rendering the basic material differences required for bone burring tasks. The current implementation, directly operating on a voxel discretization of patient-specific 3D CT and MR imaging data, is efficient enough to provide real–time haptic and visual feedback on a low–end multi–processing PC platform.

Acknowledgments

I wish to express my gratitude to my advisors, Bruno Picasso, Gianluigi Zanetti and Enrico Gobbetti, for their inspiration and guidance, and for making this whole experience possible.

I am also grateful to Antonio Zorcolo, Andrea Giachetti, and Gavin Brelstaff, CRS4 researchers, for all their help and expertise in the practical aspects of this research.

I also wish to thank Prof. Stefano Sellari Franceschini and Prof. Alistair Pettigrew, for the expertise in the medical aspects of this work, Fabio Bettio and Giovanni Pintore, CRS4 Visualization, Production, and Post-production Laboratory, for the realization of all multimedia productions related to the project, as well as DIMECA-UniCa and CRS4 technicians, for their help with the experimental equipment.

This work was supported by CRS4, that let me have the chance to attend the Ph.D. program. It would have been difficult to complete the work without this support.

Finally, I must express my appreciation to my girlfriend, my relatives, my colleagues and my friends, a source of constant encouragement. Thanks for existing.

Contents

1. Introduction

1.1. Introduction

The increasing importance of minimally invasive operational techniques is preparing the ground for a drastic change in pre-operation planning and basic surgeon training. In fact, the spatial limitation of the operational theater, the way the latter is seen by the surgeon, e.g., by microscope, and the absence of direct physical contact between the surgeon and the patient body, make it possible to replace the patient, for training and pre-operation planning purposes, with a computer system able to reproduce the right visual and tactile feed-back. In this dissertation we describe our first result in the development of a training system for simulating surgery on the temporal bone, a skull region just behind the ear. The specific target of our simulator is mastoidectomy, a very common operative procedure that consists in the removal by burring of the mastoid portion of the temporal bone. The site anatomy is widely variant. The main risks are related to the detection and avoidance of the facial nerve and of aberrant jugular veins (or branches) and to the resection of adequate amounts of the mastoid air cells. The ability to rehearse the procedure using patient specific data is extremely rare. A VR simulator realistically mimicking a patient-specific operating environment addresses this shortcoming. A number of groups are working toward this goal [SWB$^+$02, SWB$^+$02].

Our system is designed to work on patient-specific volumetric object models directly derived from 3D CT and MRI images, and to provide realistic visual and haptic feedback, including secondary effects such as the obscuring of the operational site due to the accumulation of bone dust and other burring debris. It is expected that the ability of using directly patient specific data as input will help in the accumulation of a large number of training cases. Moreover, it will open the road toward the use of the simulator for pre-operation planning and rehearsal, making thus possible to plan surgery directly on a model of the individual patient, rather than by referring to a model surgical procedure on a standard anatomy. The need to provide real time feedback to users, while simulating burring and related secondary effects, imposes stringent constraints on the system. Our solution is based on a volumetric representation of the scene, and it harnesses the locality of the physical system evolution to model the system as a collection of loosely coupled components running in parallel on a multi-processor PC platform.

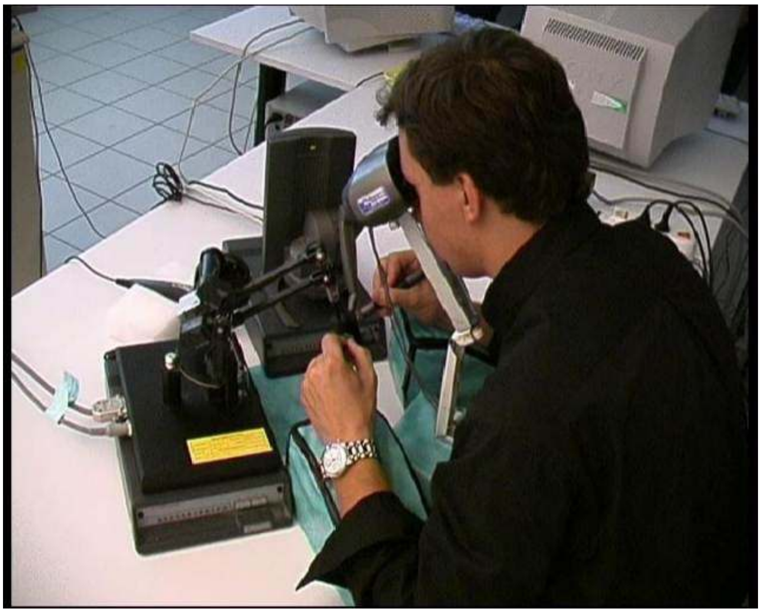


Figure 1.1.: **Surgical Simulator Setup.**

Figure 1.1 shows the current configuration of our virtual reality training system for temporal bone surgery. It is composed by two phantom haptic devices that provide force feedback for sucker and burr, as well as an N-vision binocular display that presents images to the user.

Figure 1.2 shows the principal steps of a virtual basic mastoidectomy, performed by an expert surgeon.

The prototype described in this dissertation demonstrates that this option is viable and subjective input from selected end users is encouraging.

1.2. Background and Motivation

Modern surgery is in a period of heavy transformations, due to the following causes:

- **Strong diffusion of minimally invasive surgical techniques.** The heavy diffusion of these approaches has lead to great improvements in the quality of interventions, but with the cost of

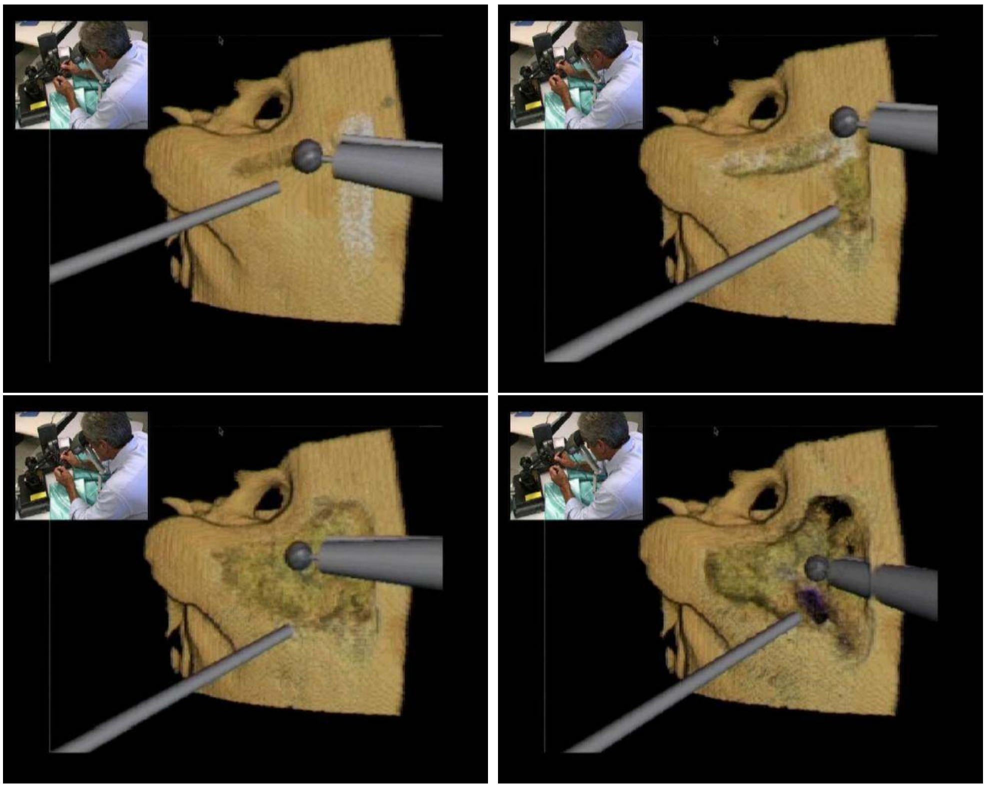


Figure 1.2.: **Sequence recorded from the simulator**: the principal steps of a basic mastoidectomy, performed by an expert surgeon are represented. Thanks to Prof. Stefano Sellari Franceschini, University of Pisa, for performing the virtual operation.

a high increase of the technical complexity and a consequent increase of time required for training and learning [Fuc02].

- **Decreasing of training time available.** Medical technologies evolution time is continuously diminishing, and it is now arrived to be shorter than average surgeon professional life. As consequence, it arises, and it is becoming ever more important, the need to attend training courses, even for full working surgeons. This fact obviously strongly constraints the available time for training activities.

- **Changes in training activities** Traditionally, training surgeons learnt on cadavers, then on animals, finally on real patients. Because of a series of reasons, such as a substantial change in public opinion sensitivity, this training approach is becoming impracticable. In italian specific case, for example, laws do not permit to employ cadavers for training practice, excepted that coming from previous specific “in vita” donations.

- **Increase of surgeon legal responsibility.** In present time, the public perception of doctors and surgeons activities is deeply changed. In past, the surgeon was considered a “miracle man”, that could save the patient life, and he was allowed to fault, but now, is ever more considered as a professional that, implicitly, is working under a contract. So, patients or their relatives now tend to take legal actions, if they are not fully satisfied of interventions outcomes.

- **Need to certificate the professional activity.** The need to guarantee surgical quality standards is a reflex of the medical *professionalism*. In this way, patients have the instruments to judge and to compare between surgeons. Hence, it is also important to take into account the impact of surgical malpractice over public opinion.

In this sense, the medical community strongly approves [Sat96, Fow01, L.02] that, in order to better handle this deep evolution, a drastic change in training techniques and certifications will be needed. The usage of surgical simulators based upon virtual reality technologies promises to be the principal actor in this evolution. In fact, analogously with aerospatial technology, where simulator are currently used as fundamental training and certification instruments, the employment of surgical simulation technology will allow:

- a great flexibility in training sessions;
- to gradually modify the training difficulties;
- to expose trainees to rare events, that can be very dangerous for patient life;
- to quantify *performance* and surgical skills.

The work described in this dissertation contributes to progress in this direction, as we will show in following sections and chapters. The principal contribution of this work is the haptic simulation of the contact between surgical burr and biological material. In this chapter we will introduce the objectives of this dissertation, and the organization of the chapters.

1.3. Objectives

The principal objective of this work is to show that it is possible to render realistic haptic sensations of the contact between surgical burr and biological material, and that this kind of sensations are useful for the realization of virtual reality based training systems. The kernel of the thesis is the derivation of simple physically based mathematical models that can replicate the behavior of a spherical rotating burr during typical surgical tasks. These models rely on a few parameters, and with the realization and the employment of an experimental facility, we want to empirically derive the values of these parameters for materials of interest. The experimental results can be compared with domain expert indications, and they will show whether subjective tuning provides values close to those derived from experimental calibration. For this reason, in cases where experimental calibration is not practice-able or too complex, we want to show that the employment of subjective calibration techniques is to be considered a correct way for tuning haptic simulations. Finally, an intensive series of psychophysical testing will show whether our physically-based haptic model of a surgical burr is capable of rendering different materials, and what are the principal haptic cues in surgical burring. Moreover, this dissertation describes a virtual reality training system containing the haptic models, and provides details about the design and implementation, showing that the realistic immersing sensations are obtained by the employment of accurate visual volume rendering as well as realistic physically-based visual secondary effected. In order to obtain immersing sensations timing requirements needs to be satisfied, and this fact impose constraints in the system design. The existing prototype is currently under evaluation, and in this dissertation we will provide preliminary results, obtained during virtual training sessions of ENT surgeon trainees.

1.4. Achievements

The principal achievements obtained during the PhD work are the following:

- physically-based haptic models of the behavior of a spherical surgical burr during bone dissection tasks;
- a “custom” experimental facility, used to characterize the behavior of materials of interest for surgical burring, including temporal bone specimens, and plastic replies commonly used for training purposes;

- the design and implementation of a surgical training system prototype containing the implementation of the haptic model of the cutting burr, an hardware-accelerated volume renderer, as well as a simulation of the visual secondary effects involved in the surgical specialty considered;
- the design and implementation of psychophysical experiments, to investigate the tactile cues used in distinguishing real materials, and related results.

1.5. Organization

The dissertation is organized as follows:

- Chapter 2 introduces and describes the problems related to drastic changes in modern surgery, and the consequential revolution in training techniques. In this context, Virtual Environments promise to be the principal actors to manage these changes.
- Chapter 3 describes the surgical specialty chosen to benchmark our surgical simulator prototype. We decided to study temporal bone surgery, because of its diffusion and complexity.
- Chapter 4 provides the task analysis involved in the project, to find the user requirements for the surgical simulator.
- Chapter 5 introduces the problems related to the generation of haptic stimuli for the simulation of surgical burring. It also provides a mechanical description of the haptic interfaces currently employed.
- Chapter 6 discusses the haptic implementation of a bone cutting burr, that it is being developed as a component of our training system for temporal bone surgery.
- Chapter 7 describes the adaptive implementation of the model presented in chapter 6. The adaptive technique exploits a multiresolution representation of the first two moments of the bone characteristic function to rapidly compute contact forces and determine bone erosion.
- Chapter 8 describes the strategy followed to collect experimental data that will be used to develop and validate our physically based contact and bone erosion model. The experimental set–up consists of a force–controlled robot arm holding a high–speed rotating tool and a contact force measuring apparatus.

- Chapter 9 discusses the rendering technique implemented inside the surgical simulator, that enables to render dynamically changing volumes. The approach is based upon texture mapping hardware and the NVIdia OpenGL combiner extensions.
- Chapter 10 describes the simulation visual component, used to simulate secondary obscuring effects, such as bone debris accumulation, blooding, irrigation, and suction. The approach uses a time-critical particle system evolution method.
- Chapter 11 provides the details about the system design and architecture. The current implementation is also described.
- Chapter 12 demonstrates how expert surgeons and trainees can effectively use the system for training and assessment purposes. Preliminary kinematic and dynamic analysis of simulated mastoidectomy sessions are presented.
- Chapter 13 provides the results on the tuning of the temporal bone surgical simulator using parameter values derived from experimental measurements. It also describes results obtained by psycho-physical testing.

Part I.

Background and Motivation

The specific target of our efforts in surgical simulation is temporal bone surgery. Mastoidectomy is the most superficial and common surgery of the temporal bone. It consists of removal of the air cavities just under the skin behind the ear itself, and it is performed for chronic infection of the mastoid air cells (mastoiditis). This part of the dissertation introduces the problems related to computer–assisted surgical simulation. Specifically, chapter 2 describes the future role of virtual environments for surgical training, chapter 3 introduces the application area of our surgical simulation system, and chapter 4 reports on the task analysis carried out in order to identify the features relevant to a training simulator for temporal bone surgery.

2. Motivation: Surgical Training

This chapter shortly describes the problems related to new surgical trends and their effects on training. In this scenario virtual environments promise to be principal actors. In fact, training in virtual environments has many potential advantages. It is interactive and changes can be made that demonstrate variations in anatomy or disease state. Simulated positions and forces can be recorded to compare with established performance metrics for assessment and credentialing. Students could also try different techniques and look at the tissues from perspectives that would be impossible during a real operation.

2.1. Introduction

Training in surgery is principally based on an apprenticeship model. Residents learn by watching and participating, taking more active roles in the operation as their experience increases. This model has survived in part because of the limitations of the training media available outside the operating room for teaching surgical skills, and in part because the techniques of traditional open surgery mostly rely on familiar eye-hand coordination, allowing most residents to achieve competence by repeated practice. Two events are occurring that may lead to a significant change in the nature of surgical training. First, increasing numbers of surgical procedures are performed using minimally invasive techniques, in which trauma to external tissue is minimized. The skills of minimally invasive surgery (MIS) present unique perceptual-motor relationships that make these skills very difficult to master. Second, virtual environments with the capability to teach surgical skills are becoming available. Training in virtual environments has many potential advantages. It is interactive, yet an instructor presence is not necessary, so students can practice in their free moments. Changes can be made that demonstrate variations in anatomy or disease state. Simulated positions and forces can be recorded to compare with established performance metrics for assessment and credentialing. Students could also try different techniques and look at the tissues from perspectives that would be impossible during a real operation. During the present chapter we describe the surgical training traditional methods with their actual defects. We then show how training can be improved through the aid of virtual reality based simulators.

2.2. Traditional surgical training

The contemporary junior surgeon develops his skill through literature, textbooks, lectures, observation, and ultimately by performing the procedure under the supervision of an experienced surgeon. Due to the nature of this progression, the quality of education is still quite unpredictable and mainly depends on the instructor and the particular cases to which the surgeon is exposed during his or her training. There are studies showing that the outcome of surgery is significantly worse on the first procedures performed by an inexperienced surgeon [DC95]. The management of many of the complications and variations which arise during a procedure cannot presently be taught as this would put the patients at an unacceptable risk. The need for a comparable assessment of the physician abilities and competence predicates even more problems, especially in surgical education. In following subsections the traditional training approaches are briefly explained and reviewed.

2.2.1. Apprenticeship model

Surgical residents still learn primarily through the apprenticeship model, wherein a novice assists a surgeon with greater experience in a particular procedure. Simply watching a fully trained surgeon performing a procedure does not allow one to develop a clear understanding of the relationship between the anatomic structures encountered during the operation. The student needs hands-on experience with examining and manipulating the organs and the surgical instruments used during the procedure in order to become proficient.

2.2.2. Live animal laboratories

Students have traditionally learned the basics of surgery in live animal laboratories. Learning to dissect and use surgical instruments begins early in training with anatomy lab dissections and physiology lab experiments. In graduate or professional school, surgery classes using animals, usually dogs, have typically been part of the curriculum; for example, an initial exercise might involve anesthesia of the animal, a skin incision, hemostasis, and closing the wound with sutures. The consensus has historically been that such practice on live animals is necessary. More recently, however, the use of live animals for surgical training has been controversial. Criticisms have included the prodigious expense of such labs, the inappropriate use of animals, and inferiority to other teaching methods. Currently, there is an evolving system employing multiple approaches, including both animal laboratories and inanimate training aids. Live animal surgical labs have distinct advantages, including authenticity and proven efficacy. Obviously, using live animals is the most realistic training method, exposing students to a system with respiration, blood circulation, peristalsis, and other working bodily functions. The experience gained in such laboratories can help overcome hesitancy and timidity. Further, as the traditional training method, live animal laboratories have a proven track record. On the other hand, this training method also has distinct limitations. For many operations, there is only one opportunity to perform the procedure. If a student makes a mistake, he or she will most likely be unable to begin again. This inefficiency is part of the reason it takes five to eight years to train a surgeon. Because students are typically organized into surgical teams of up to four students operating on a single animal, only one or two of the students actually performs the surgery. The remaining team members, who are not scrubbed, might learn valuable support functions, such as anesthesia and animal maintenance, but are not developing surgical skills. While these other functions are also important skills, faculty with expertise in these areas may teach them more effectively in the appropriate setting. Surgical technicians and research scientists who use animals in their studies and are from non-medical backgrounds are sometimes trained without the benefit of a comprehensive course. A mentor or superior may simply show them a particular operation necessary to accomplish a research objective, which they practice until learned. Basic education in aseptic technique is often not provided. Problems may be passed on when one student becomes the teacher of the next. Cost was reported as the main reason for discontinuing animal laboratories. Expenses include cost of space and equipment to furnish a surgical station, animal procurement and care, and surgical supplies. Additional expense is incurred if faculty time is not used efficiently. Because inanimate training aids are more cost-effective, most courses using animals also used dry labs employing inanimate trainers or cadavers [MBD99].

2.3. Cadaver laboratories

A cadaver can provide the added dimension of learning surgical anatomy at the same time. There is evidence that students training on cadavers develop proficiency equal to those using live animals. The problem in using cadavers, as reader can imagine, is related to public opinion sensitivity. Currently, cadaver laboratories are considered illegal in the majority of nations, and this kind of approach is becoming even more rare.

2.4. Modern surgical trends

Surgery is currently in a period of deep transformation, due especially to the rise of new approaches. **Minimally invasive surgery (MIS)**, or ”keyhole surgery” as it is sometimes called, has caused a revolution in surgical practice and technology. Traditional surgical approaches have utilized relatively large incisions designed to provide optimal visibility and exposure of the operation site. MIS, on the other hand, utilizes small incisions measuring a few centimeters or less, resulting in fewer traumas for the patient and yielding significant cost savings as a result of shorter hospitalization times and reduced therapy requirements. Other benefits of minimally invasive surgery are less pain, less need for post-surgical pain medication, less scarring and less likelihood of complications related to the incision. In MIS procedures, a miniature camera is introduced into the body through a small incision and transmits images back to a video monitor, enabling the physician to diagnose and if necessary treat a variety of conditions. To treat the condition, the physician inserts surgical instruments and auxiliary devices such as irrigation/drainage devices through one or more small incisions.

2.5. Role of Virtual Environments in modern surgery

Virtual reality promises to change the world of surgical training and practice. Just as flight simulators revolutionized pilot training, human simulators will become the medical classrooms of the future. Surgeons will be able to train on simulated human models, perfecting their techniques without even entering an operating room. Recent advances in computer technology have placed these exciting ideas within our reach. The role of VEs in modern surgical training is strictly related to the following points [KRL+03]:

1. Surgical techniques have become increasingly complex, thus making the learning curve to master these techniques steeper and longer.

2. More complex intervention techniques are rapidly developed and introduced in the daily practice.

3. The conventional surgical teaching method is a close daily working relation between the experienced teacher (trainer) and the unskilled pupil (trainee).

4. In traditional teaching the steep learning curve takes place during the interaction with real patients.

5. The modern patient does not accept any mutilation attributed neither to the disease nor to the intervention.

6. It is clear that a perfect preoperative visualization and planning, and rehearsals of these interventions are essential.

7. This means that while there is an increased demand for surgical training, experienced surgeons have increasingly less time and opportunity to cope with this demand. A dedicated medical VE is badly needed to lift this burden from their shoulders.

8. It is also of paramount importance to allow trainees to explore critical situations and to let them experiment with an underlying model of the phenomena and processes in the human body, without the stress of having to deal with an actual patient.

9. Virtual surgical tools should be available for life-long medical education and assessment of the surgical consultant.

A virtual reality surgical simulator could offer the possibility of having the surgical resident of the future perfect a procedure without harming a patient, learning surgical anatomy and repeatedly practicing technique prior to performing surgery on the actual patient. This would translate into a very objective exam for certification using the exact same machines. With the beginning of the twentieth century, flight simulators were introduced and soon became a proven means of training pilots in complex maneuvers [Hab86, MJ86]. Flight simulators provide an environment for learning and instruction, a tool for prediction, and an aid for experimentation. Their advantages include decreased costs and increased safety compared to real flight experience. The first simulators provided only a very vague representation of reality, using vector graphics without any texture and a very low number of picture-frames per minute. But with advancements in computer technology, these simulators have become a good deal more complex and realistic. So much so that telling the difference between reality and simulation is sometimes impossible. For the most part, the advantages of flight simulators hold equally true for surgical simulation [Sat95, L.02]. Surgical simulators provide a concentrated environment that lends itself to learning complex tactile maneuvers in a relatively quick and proficient manner. Moreover, simulation of infrequent but highly hazardous events provides experience in handling these scenarios that may not be available during a period of routine flight or surgical training. Like flight simulation, surgical simulators allow the user to train to perform a complex task using an interactive computer environment. Over the last century, this interactive environment has progressed from a two-dimensional screen (i.e. photographs and radiographers) to a three-dimensional virtual reality. Two-dimensional sources of data were initially modified by hand using drafting tools. This two-dimensional data was subsequently introduced to a computer in order to facilitate manipulation and give the surgeon the ability to better plan and demonstrate the outcome of the proposed procedure. More recently, volumetric data obtained from computer-aided scans have provided three-dimensional information for the surgeon to assist in planning complex operations. Using a computer simulator for planning, a surgeon may try out many possible reconstructions on a patient-specific model prior to operating. Surgical simulators consist of three basic components similar to those of a flight simulator: the computer, the interface, and the physical model. The physical model for the surgical simulator is a realistic computational representation of the patient, the operating room, and the surgical instruments [Fol87, PV95, SZP89]. The interface uses either a mouse or glove so the user can manipulate surgical instruments three-dimensionally, and it uses internal motors to give the user a sensation of force-feedback. Through this feature, the user can move a scalpel into the virtual tissue and actually feel its resistance, all simulated according to real patient information.

2.6. Conclusion and Discussion

Because the systems for virtual reality are improving so rapidly and because this new technology is quickly moving into the operating room, we must reassess the role of VR in surgical training and planning. The systems and possibilities discussed above are only the beginnings of fascinating future technology and its potential use in medicine. As indicated, the biggest hurdle we face today is designing an improved model of the human body for VR surgery. Further work also needs to be done on the tools used to interact with this model. Despite its centrality in the medical field, the virtual human has practical applications in areas like transportation for crash testing, the military for ballistics research on tissue injury, and commerce for ergonomic design studies. Even though years of experience have proven that most aspects of surgical training can only be learned by exposure to real patients in real physical environments, there are other things that can be more easily learned on VR simulators available today, such as perfecting manual skills and treating rare disorders. Furthermore the VR systems introduce the alluring possibility of a completely objective measurement and assessment of the trainee ability. For more advanced tasks, a robust model of the human body is needed to aid in the planning of surgery. More work is needed to refine this particular computer model and validate its results. The future will have computers not only involved in the training of surgeons, but also in the planning of surgery and the aiding of performance in the operating room. Ultimately the acceptance of these simulators and trainers depends heavily on the realism of the virtual human body models on which they are based.

3. Application Area: Temporal Bone Surgery

This chapter introduces the application area of our surgical simulator. The specialty considered is the skull base surgery, and specially the mastoidectomy, that is the most superficial and common surgery of the temporal bone. It consists of removal of the air cavities just under the skin behind the ear itself, and it is performed for chronic infection of the mastoid air cells (mastoiditis). The mastoid air cells are widely variant in their anatomy and the main risks of the procedure are related to the detection and avoidance of the facial nerve, venous sinuses and "dura madre".

3.1. Introduction

The application area of our work on surgical simulation is the temporal bone surgery. Specifically, the target of our simulator is mastoidectomy, a very common operative procedure that consists in the removal by burring of the mastoid portion of the temporal bone. The site anatomy is widely variant and the main risks are related to the detection and avoidance of the facial nerve and of aberrant jugular veins (or branches) and to the resection of adequate amounts of the mastoid air cells. In the typical mastoidectomy surgical setup, the Ear Nose and Throat surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker, that he uses, respectively, to cut the bone and to remove water (used to cool the burr bit) and bone paste produced by the mixing of bone dust with water. The chapter is organized as follows: section 3.2 provides informations about the temporal bone anatomy, section 3.3 reviews the typical surgical specialties regarding the temporal bone, while section 3.4 focuses on the surgical equipment. Finally section 3.5 describes the principal difficulties in such surgical specialties and section 3.6 reviews the current available training aids.

3.2. Anatomy of the temporal bone

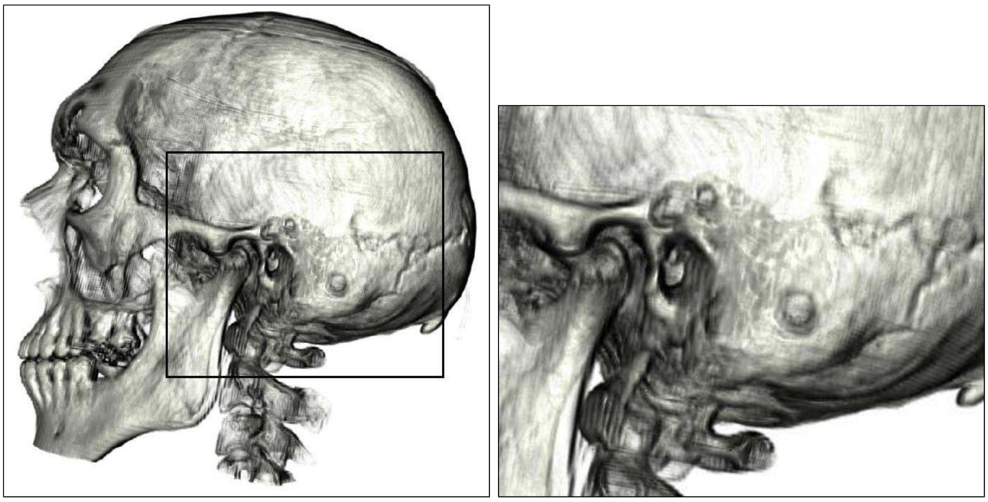


Figure 3.1.: **Mastoid anatomy**:the temporal bone forms the osseous encasement for the middle and inner ear.

The temporal bone (figure 3.1), which forms the osseous encasement for the middle and inner ear, consists of three parts: squamous, tympanic and petrous. The squamous part is broad in shape, thin and flat. The tympanic part is interposed below the squamous and anterior to the petrous parts. The external acoustic meatus, which leads to the tympanic membrane, is surrounded by the tympanic part of the temporal bone. The hard petrous part contains the organ of hearing and the vestibular canals. Its mastoid process is not solid but contains many air cells. Its external surface affords attachment to several muscles. The tympanic cavity (**middle ear**) communicates posteriorly with the mastoid antrum which, in turn, connects with the mastoid air cells. The lateral wall of the tympanic cavity is formed principally by the tympanic membrane, while the medial or labyrinthine wall presents several structures of importance. These structures include: the vestibular window (oval window), the cochlear window (round window), the promontory (projection of the first turn of the cochlea), the prominence formed by the facial canal and, more posteriorly, the prominence formed by the lateral semicircular canal [Chi00]. Figure 3.2 shows a panoramic view of the mastoid region produced by our surgical simulator viewer. The volumetric dataset is 256x256x128 and it is derived from CT images.

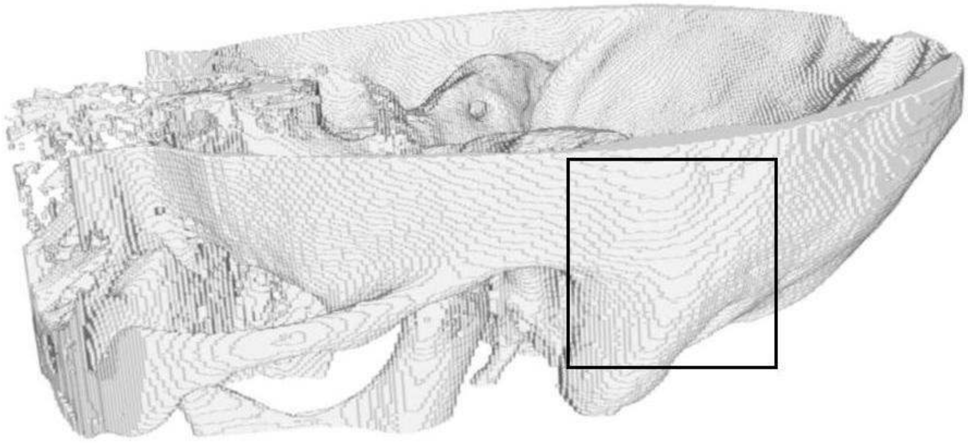


Figure 3.2.: **Surgical site.** Mastoidectomy is performed in the region indicated by the rectangle. CT data courtesy of dr. Emanuele Neri, Division of Diagnostic and Interventional Radiology, University of Pisa.

3.3. Middle-Inner Ear Surgery Specialties

3.3.1. Mastoidectomy

Mastoidectomy is an operation that allows the exposure of the mastoid air cells, middle ear space and ossicles. It is useful in eradicating chronic infections of the ear and the removal of cholesteatomas. Various forms have been proposed over the years, along with their indications and contraindications. This operation is also useful in exposing the facial nerve and in certain approaches to the inner ear structures. The figure 3.3 shows the external auditory canal (EAC) and the mastoid antrum (the largest air cell in the mastoid). The sigmoid sinus is a large vein that drains blood from the brain. The tegmen is the bone that separates the mastoid from the brain. In this subsection the typical procedure is described. Drilling should start at **Macewen's triangle** (overlying the mastoid antrum). This area is marked by drilling a straight line along the temporal line up to estimated area of the sinodural angle and a perpendicular line behind the external auditory canal up to the mastoid tip, These two lines mark the superior and anterior limits of dissection. Wide cortical removal of the mastoid is achieved by a systematic saucerization of the cavity, taking note of the tegmen tympani superiorly and the posterior canal wall anteriorly. The posterior canal wall should be thinned out. If the shadow of an instrument in the middle ear could be seen from the antrum through the posterior canal wall, the bone has thinned out well. The dural plate of the middle fossa(**tegmen tympani**) is identi-

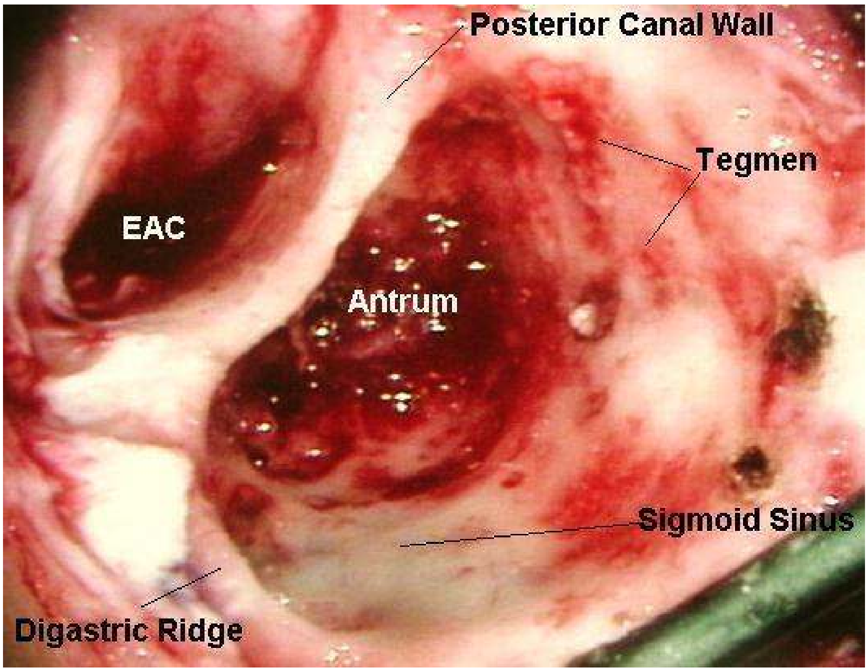


Figure 3.3.: **A typical image of the mastoid**. The external auditory canal (EAC) and the mastoid antrum are highlighted.

fied superiorly. Mastoidectomy should proceed from one level to the next. The **Koerner's septum** separates the superficial mastoid cortex (**squamous**) from the deeper mastoid cortex (**petrous**). In a live specimen, the surgeon can tell if the drilling is near the middle fossa dural plate or the lateral sinus by the bone color, that changes from white to a pinkish or bluish hue. Adequate exposure of the tegmen tympani is necessary to facilitate the location of the lateral sinus (**sigmoid sinus**) and the sinodural angle. All air cells are evacuated from these areas. The bone at the zygomatic root overlying the antrum is further drilled out to gain access to the fossa incudis. The incus can be seen at this area. With further drilling in this area, the posterior and horizontal semicircular canals, as well as the facial recess, are identified. After performing a simple mastoidectomy and posterior tympanotomy, the outline of the facial nerve can be observed [Ant, Nel91].

3.3.2. Tympanoplasty

Tympanoplasty basically means the surgical reconstruction of the tympanic membrane and the establishment of ossicular continuity from the tympanic membrane to the oval window. Taken separately, the surgical restoration of the tympanic membrane is termed **myringoplasty**, whereas the establishment of ossicular continuity is called

ossiculoplasty.

3.3.3. Inner ear surgery

An acoustic neuroma (**neurilemmoma**) is a benign encapsulated tumor that, depending on size, can produces symptoms which result from the pressure applied on other surrounding structures, such as the acoustic nerve or brain stem. There are a number of procedures available to surgeons to remove these tumors, although the choice of which to apply depends on such factors as the presence of residual hearing, the location of the tumor and the experience and preference of the surgeon.

3.4. Surgical Equipment

In the typical mastoidectomy surgical setup, see fig. 3.4, the Ear Nose and Throat surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker, that he uses, respectively, to cut the bone and to remove water (used to cool the burr bit) and bone paste produced by the mixing of bone dust with water.

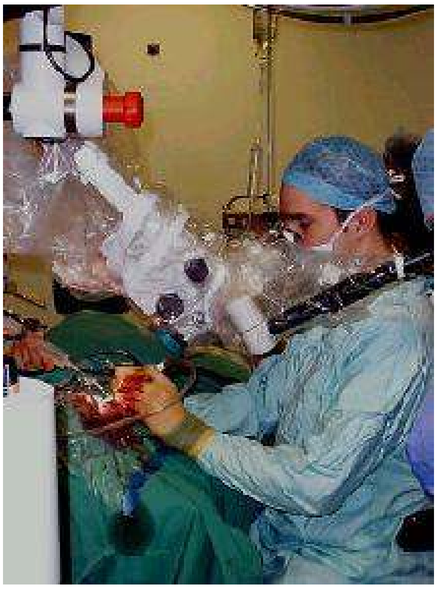


Figure 3.4.: **Middle-ear surgery equipment**. The surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker. Photo courtesy of Prof. Bob Stone, Virtual Presence Ltd.

3.5. Required skills and related problems

The mastoid and the temporal bone are widely variant in their anatomy and the main risks of the surgical procedures are related to the detection and avoidance of the facial nerve, venous sinuses and ”dura madre”. A number of basic and common task elements/surgical behaviors can be noted during surgical observations. These are:

- deep drill positioning,
- sensitive structure visual avoidance (sometimes with the added protection of an additional instrument, forming a barrier between drill and structure),
- thin bone hooking (whereby the ends of 90o dissectors were carefully tucked under the translucent thin bone and then forcibly withdrawn causing bone plate breakage),
- bone structure contour following (primarily haptic, but strong ridge shadows were available for visual accuracy and fine predictive tracking),
- simple linear tracking,
- deep drilling under conditions of partial visual obscuration of the smallest of burrs,
- reliance on certain auditory cues.

3.6. Training Aids

Figure 3.5.: A Pettigrew Plastic Temporal Bone sample.

The limitations of human cadaver dissection for teaching has led to the development of alternative training aids. Widely used are plastic models, such as Pettigrew Plastic Temporal Bone series (see figure 3.5). Using Pettigrew's models the complete temporal bone, for example, can be fully dissected using standard theater equipment [Pet].

While such synthetic models are currently a valid alternative to cadaveric exercises, simulators allow increased exposure to pathological variance through patient specific and synthesized models, and support a more quantitative assessment of trainee proficiency [KJM$^+$97]. A number of groups are thus developing simulators for bone dissection. Early systems (e.g., [HIT88]) focused on increasing the understanding of the anatomy by providing specialized visualization tools of static models. The VrTool [KJM$^+$97] and the VOXEL-MAN system [PPT$^+$02a] mainly concentrate on accurate visual presentation of free-form volume-sculpting operations. The Ohio Virtual Temporal Bone Dissection simulator [WBS$^+$00, BSWS01, SWB$^+$02], similarly to our work, aims instead at realistically mimicking the visual and haptic effects of a real operation. Throughout this dissertation, we will describe our work, that is characterized a the physically based contact model [AGG$^+$02c], the visual simulation of bone dust, irrigation, and bleeding, as well as the use of multi-resolution and time-critical techniques to meet performance constraints.

3.7. Bibliographical notes

The information and details contained in this chapter are principally derived from [Nel91], that is the fundamental reference for all surgeons that practice temporal bone dissections. Other informations about the temporal bone anatomy can be found in [Chi00], while [Ant] gives details about the surgical instruments commonly used during the bone dissection.

4. System Requirements

An extensive task analysis has been carried out in order to identify the features relevant to a training simulator for temporal bone surgery. The analysis involved a review of existing documentation, training aids, and video recordings, interviews with experienced operators, as well as direct observation of the procedure being performed in theater. This chapter reviews the procedures and results obtained by this task analysis.

4.1. Introduction

An extensive task analysis has been carried out in order to identify the features relevant to a training simulator. The analysis involved a review of existing documentation, training aids, and video recordings, interviews with experienced operators, as well as direct observation of the procedure being performed in theater. The analysis indicates that the main processes to simulate are the following: a burr reducing bone in fine dust, an irrigator introducing water that mixes with bone dust, and a sucker removing the mixture, [JTP$^+$01, JJT$^+$ar, AGG$^+$02a]. The capability of replicating the effects caused by the intertwining of the different physical processes is of primary importance for training. The absence of these effects would reduce the importance placed by a trainee on the need for regular irrigation and suction. Subjective analysis of video records, together with in-situ observations highlighted a correlation between drilling behaviors and type and depth of bone. High-quality force feedback is thus needed for the dominant hand (controlling the burr/irrigator), while only collision detection is required for the non-dominant one (controlling the sucker). As for the nature of the technology required for displaying drill, drill site, bone, and so on, binocular viewing systems are deployed in the operating theater and used by surgeons, and so binocular imaging should be available to the simulator. However, the wearing of any form of stereoscopic display, such as a head-mounted display or liquid crystal shutter glasses should be avoided.

The chapter is organized as follows: section 4.2 introduces the principles of human centered design, while section 4.3 explains the technique used for the design of our training system. Section 4.4 provides details on task analysis sessions, and finally section 4.5 derives the training environments requirements.

4.2. Principles of Human Factors Task Analysis

A task analysis is a process by which one can formally describe the interactions between a human operator and his/her working environment (including special-purpose tools or instruments), at a level appropriate to a pre-defined end goal (typically the evaluation of an existing system or the definition of the functional and ergonomic features of a new system) [May99]. Without a properly executed task analysis, one runs the risk of specifying or designing a VR (or any computer-based training or multimedia) system that fails to record or measure those elements of human skill one was targeting in the first place. The task analysis should form an early and central component of any project that involves a major human-centered component. Indeed, recognition of this has recently been formalized by the publication of International Standard ISO 13407, Human-Centered Design Processes for Interactive Systems (ISO, 1999) [JIMK03]. ISO 13407 specifies 4 general principles of human-centered design and 4 further principles of human-centered design activities. Namely the principles of human-centered design are:

1. Ensure active involvement of users and a clear understanding of user and task requirements (including context of use and how users might work with any future system evolving from the project if at all);

2. Allocate functions between users and technology (recognizing that todays technology, rather than de-skilling users, can actually extend their capabilities into new applications and skill domains);

3. Ensure iteration of design solutions (by involving users at as many stages of the design and implementation process as is reasonable practical),

4. Ensure the design is the result of a multidisciplinary input (again this emphasizes the importance of user feedback, but also stresses the need for input from such disciplines as marketing, ergonomics, software engineering, technical authors, etc, etc).

while the four human centered design activities are:

1. Understand and specify the context of use (including the characteristics of the intended users; the tasks the users perform, or are to perform; the environment in which users use, or are to use the system; relevant characteristics of the physical environment);

2. Specify user and organizational requirements (in the context of the present project, this includes aspects of team working, health and safety issues, user reporting structures and responsibilities);

3. Produce design solutions (with multidisciplinary team and user involvement);

4. Evaluate designs against requirements (a continuous process throughout the design cycle).

This methodology has been closely followed for the task analysis invloved during the design of the simulation platform for temporal bone surgery training.

4.3. Surgical Training System Design

There are many techniques for carrying out a task analysis. Some involve observational and/or interview techniques (often backed up with video and/or audio records). Others can employ quite sophisticated computer-based solutions, from mixed media data recording (video, keystrokes, physiological parameters, voice, etc.) to simulations based on human performance parameters and cognitive models. Due to the short time available for the present analysis sessions (and the need to fit in with the schedules for specific operations), a combination of interface survey and time-line analysis was chosen, allowing for the following types of raw data to be recorded:

1. off-line, through documentation, interview and personal experience with existing training techniques (cadaveric temporal bone, CD-ROM training and synthetic skull dissections),

2. in-theater observation and verbalization of and from practicing surgeons,

3. first-level review of video records of operations.

4.4. Analysis sessions

The task analysis took the form of two main processes:

- In-theater video and verbal protocol sessions,
- Initial investigation of other, synthetic training techniques (CD-ROM and plastic bones).

4.4.1. Video and verbal sessions

With the support of the staff of the ENT unit, the Department of Otolaryngology, University of Pisa, we were permitted to observe video records of surgeon performance and close-in drilling activities. Various theater sessions were analyzed:

- Infantile Cochlear Implant
- Fossa Acoustic Neuroma
- Trans-labyrinthine Acoustic Neuroma

Video records showed some trends in drilling behaviors depending on the type and depth of bone (eg. cortex vs. petrous). In the case of preparing the skull wells for the receiver/stimulator, burrs of around 0.8cm were used in conjunction with sweeping motions and curvilinear sweeps over 2-4cms coordinated by flexure and extension of the forefinger and thumb pinch grip around the drill. The mastoidectomy was characterized by similar, albeit shorter (1-2cm) motions with rapid lateral strokes. For deeper drilling, $\leq$ 1cm strokes – down to 1 or 2mm – become evident (obviously with smaller diameter cutting and diamond burrs – down to 1-2mm in size for cochleostomy) with more of a “polishing” motion quality or very sensitive, almost exploratory motions, guided using the contours from prior drill procedures. “Static” drill handling was also noted, suggesting extreme caution on the part of the surgeon to simply allow the cutter to erode the bone tissue whilst maintaining minimal surface pressure.

4.4.2. Temporal Bone Dissector CD-ROM

As well as the actual video experiences and interviews described so far, every attempt was made to investigate and evaluate other electronic or synthetic training products. The Temporal Bone Dissector CD, published by Mosby, is a highly informative ontology reference, if somewhat lacking in training quality. The CD has been developed using a combination of Macromedia animation and QuickTime movies. As with many attempts to illustrate the temporal bone and middle/inner ear contents and structures, the CDs interactive and atlas images show little relevance to what is found in the real world. This CD possesses a wealth of good introductory information. Even though the operations are rather simplistic, trainees receive feedback on their performance, in the form of brief comments during drilling processes and in the form of a happy or dissatisfied (and ready-for-litigation) cartoon patient at the end of a full session.

4.5. Training environment requirements

The term training environment refers to those aspects of the surgeons task that will need to be replicated for the purposes of a procedural and skills trainer. The key features to support include:

- **Initial Bone Exposure.** The initial incision and cuts are fairly basic, from a surgical skills standpoint, so a high-fidelity tissue simulation for this stage in the task would be cost-ineffective. Although surface marking was a standard procedure for acoustic implant planning, it is felt that the same facility could be provided in the training system to help ENT trainees plan their initial cuts for mastoidectomy operations and Middle Fossa (even Retrosigmoid) approaches. The head representation or section (including auricle) should be 3D in nature. However, stereoscopic viewing is not considered necessary for this task. There is no reason why a conventional, high-resolution display, together with the 6-/3-dof data input/haptic feedback device selected for the training system could not fulfill this role. Once the planned incision path and skin area markings were completed, the system should be capable of informing the surgeon of any gross errors between his marks and those of an optimized area (eg. mark deviations > 10mm (posterior) and 5mm anterior)). Other patient preparatory processes were undertaken during the operations observed (eg. subcutaneous abdominal fat removal for wound packing), but the skills for this type of activity are, again, well covered in the basic surgical skills courses and are not worth simulating.

- **Drilling/Burring (sweep, spiral, contour and down-pressure).** Subjective analysis of video records, as described earlier, together with in-situ observations highlighted a correlation between drilling behaviors and type and depth of bone. In the case of initial cortex burring and recess preparation for the cochlear implant receiver/stimulator, drill tip/burr motions of around 0.8

cm together with sweeps over 2-4 cms were evident, as were fine flexure and extension movements of the forefinger and thumb around the drill. Shorter (1-2 cm) motions with rapid lateral strokes characterized the post-cortex mastoidectomy. For deeper drilling, 1cm strokes down to 1 or 2mm were evident with more of a polishing motion quality, guided using the contours from prior drill procedures. Static drill handling was also noted, eroding bone tissue whilst maintaining minimal surface pressure. From a training technology viewpoint, there is little doubt that the only commercially available and viable system capable of replicating these qualities is Sensable Technologies PHANToM (desktop, 1.0, or 1.5A with stylus encoder). Note also that in some cases the pressure applied to the drill site was sufficient to move the head of the anesthetized adult patients a good 5-10cms. As it has not been possible to record actual exerted forces, representative users (as demanded in ISO 13407) must be employed to act as perceptual judges in the construction of a haptic sensation library for key temporal bone tasks. Other relevant haptic cues will be considered later. Visually, the actual drill representation need only be quite simple, and it is felt that representing the spinning of the cutter or diamond burr is unnecessary. What is necessary, from a functional standpoint, is an effective collision detection mechanism which not only copes with increased resolution as the virtual drill proceeds deeper into the temporal bone, but is also capable of generating error states when (for example) a large burr is inserted into a narrow drill site. As for the nature of the technology required for displaying drill, drill site, bone, and so on, there is no conclusive evidence or support for the premise that the use of a stereoscopic system will aid performance in this case. If the simulation achieves a reasonable level of fidelity, then the combination of high-resolution images and haptic feedback will, more than likely, suffice. As well as the visual and 6-dof input/3-dof haptic feedback qualities of the PHANToM for drill simulation (including high frequency vibration), the training system would, it is felt, be enhanced by the inclusion of sound. Some surgeons suggest that they are able to detect subtle changes in sound depending on the nature of the bone they are working with (eg. cortex vs. petrous).

- **Other Instruments and Materials.** Many instruments and materials were deployed during the course of the observed operations. Only those considered of major importance to a temporal bone training system are covered here. The main visual and/or haptic representations should include:

  - Instruments used for hooking (eg. thin bone), probing (eg. manual testing for dura or bone) and peeling (eg. neuroma tissue): visual and haptic representations,
  - Instruments used for special purposes (eg. diathermy standard probe and hoop; facial stimulator), visual and haptic representations including on-screen visual representation of current / sensitivity settings (hidden until icon selected by user),
  - Materials used for soaking or coagulation (eg. Surgical mesh or swab representations); visual only,
  - Instruments used for cleaning (eg. irrigation and sucker); visual and (possibly limited) haptic representations (verbal requests for irrigation could be implemented by simple speech recognition).

  The application of haptic technologies to the use of simulated instruments other than the main drill has to be questioned from a functional and price perspective. Of those instruments listed above, the first 3 categories are actually deployed by the surgeon using his dominant hand (right hand in every case during the present analysis). The use of suction tubes was governed by his left/non-dominant hand and irrigation was provided on demand either by the theater nurse or by the integral mechanism of the diathermy system. One has to question, therefore, the need for a second PHANToM device for what is, essentially, a minor task involving suction. The only real support for using a full PHANToM system was the finding that the suction tube was occasionally used for retrieving flat bone tissue or holding dura in place.

- **Errors and Performance.** Finally, a brief mention of the need to design the virtual training environment and integrate device drivers with evaluation and human performance assessment in mind. Further effort is required to specify the scope of tasks expected of trainees. This should be kept in mind as the human-system interface develops and consultation with future users is essential to ensure the correct selection of tasks and measures. These may include overall time, reaction time to contingencies, multiple errors whilst marking out head incision and cut area, contact with key anatomical features (eg. facial nerve, jugular bulb, sigmoid sinus, brain tissue), inadequate drill/burr selection, over-pressure whilst using the PHANToM and so on.

4.6. Conclusion

An extensive task analysis has been carried out in order to identify the features relevant to a training simulator. The analysis involved a review of existing documentation, training aids, and video recordings, interviews with experienced operators, as well as direct observation of the procedure being performed in theater. The analysis indicates that the main processes to simulate are the following: a burr reducing bone in fine dust, an irrigator introducing water that mixes with bone dust, and a sucker removing the mixture, [JTP$^+$01, JJT$^+$ar, AGG$^+$02a]. The capability of replicating the effects caused by the intertwining of the different physical processes is hence considered of primary importance for training.

4.7. Bibliographical notes

The results presented in this chapter were obtained within the framework of the European Union IERAPSI project (EU-IST-1999-12175). The initial analysis sessions were performed by Prof. Bob Stone (Virtual Presence LTD.). We later extended the analysis in cooperation with Prof. Stefano Sellari Franceschini (University of Pisa). Readers interested to the human centered task analysis carried out in order to derive the user requirements for the design of the surgical simulation platform, can find additional informations in reference [Sto01]. Also reference [AGG$^+$02a] focuses on the human factor analysis and provided the basis for this chapter.

Part II.

Haptic System

The task analysis described in chapter 4 indicates that, for the replication of temporal bone surgical specialties, the main processes to simulate are the following: a burr reducing bone in fine dust, an irrigator introducing water that mixes with bone dust, and a sucker removing the mixture. High-quality force feedback is thus needed for the dominant hand (controlling the burr/irrigator), while only collision detection is required for the non-dominant one (controlling the sucker). Moreover, to be able to feed the appropriate tactile inputs to the human perceptual system, the system needs to produce stimuli at around 1 KHz. The computations needed to obtain the haptic force response can be drastically simplified, since forces can be computed by just considering a small neighborhood around the contact surfaces between surgical instruments and bones. This part of the dissertation, extensively focuses on the modeling of the haptic effects of bone burring. Specifically, chapter 5 introduces the problems related to the usage of haptic interfaces in virtual environments, chapter 6 discuss the haptic implementation of a bone cutting burr model, while chapter 7 describes adaptive techniques for providing real-time haptic feedback during the virtual bone dissection simulation. Finally, chapter 8 describes the strategy followed to collect experimental data, in order to calibrate the physically based contact and bone erosion model.

5. Haptic Devices

This chapter introduces the problems related to the usage of haptic interfaces for surgical burring. Starting from a review of the haptic devices commonly used in virtual environments, we provide a mechanical description of Sensable PHANToM haptic interfaces, that are currently employed in our surgical training system.

5.1. Introduction

As we start the 21st century, programmable and intelligent mechanical systems are becoming more prevalent in our lives. One particular area of interest is mechanical systems intended for use directly with humans; such systems share desired workspace and interact with humans to complete specific tasks. More explicitly, many of these devices consist of a mechanism an operator physically manipulates to position a particular part of the device or “end point”. The mechanical system may merely record the “end point’s” path or restrict an operator’s movements to a preprogrammed path, possibly assisting the operator’s effort. Such devices are called **haptic displays** [Bur00]. The word ”haptic” means ”of or relating to or proceeding from the sense of touch”. A haptic interface is a device which allows a user to interact with a computer by receiving tactile feed back. This feedback is achieved by applying a degree of opposing force to the user along the x, y, and z axes. Another way of thinking of a haptic display is as a super joystick that is used to operate another robot or simulate a programmed virtual environment. They resist the humans motion in certain directions, simulating barriers or constraints on where the joystick can be moved based on feedback from the teleoperated robot or programmed virtual environment [Bur00].

They have a wide range of applications, not yet deeply explored. Actually, they could be used to train physical skills such as those jobs requiring specialized hand-help tools (e.g. surgeons, astronauts, mechanics), to provide haptic-feedback modeling of three dimensional objects without a physical medium (such as automobile body designers working with clay models), or to mock-up developmental prototypes directly from CAD databases (rather than in a machine shop) [HJ95].

This chapter introduces the problems related to the usage of haptic interfaces for surgical simulation and training. Starting from a review of the haptic devices commonly used in virtual environments (section 5.2), we provide a mechanical description of Sensable PHANToM haptic interfaces (section 5.3), that are currently employed in our surgical training system. We chose Sensable PHANTom haptic interfaces, for the following pratical reasons:

- the commercial availability and cost range
- the relatively small space requirement
- the pen-like handle, similar to surgical tools.

These features make the Sensable PHANToM suitable for the haptic simulation of surgical tasks.

5.2. Survey of haptic interfaces

Haptic interfaces are devices that enable manual interaction with virtual environments or teleoperated systems. In this section, a collection of haptic interfaces for the VEs are described and appropriate description and comments about each of them is given [Hol03]. Haptic exploration is a sensory task which identifies surface or volumetric properties. There are two main parts in the human tactile sensory system: tactile information and kinesthetic information. For tactile information, mechanoreceptors in the finger pad play the major role, while for the kinesthetic information, sensory receptors in the skin around the joints, joint capsules, tendons, and muscles play the major role. Various interfaces have been built in industry and research community which use either of these two sets of information. General-purpose commercial haptic interfaces used today can be classified as ground-based devices (force reflecting joysticks and linkage-based devices), body-based devices (gloves, suits, exoskeleton devices) or tactile displays. The first two types (body-based and ground-based devices) of haptic interfaces excite human kinesthetic sensors while the third type (tactile displays) excites tactile sensors.

5.2.1. Body-based haptic devices

Passive Exoskeleton gloves

Passive gloves are devices not force reflecting the provide the means for grasping objects in virtual environments, without providing haptic sensations. The most popular commercial devices of this category are:

- Immersion CyberGrasp
- Mattel PowerGlove
- 5DT 5th Glove
- PINCH Glove

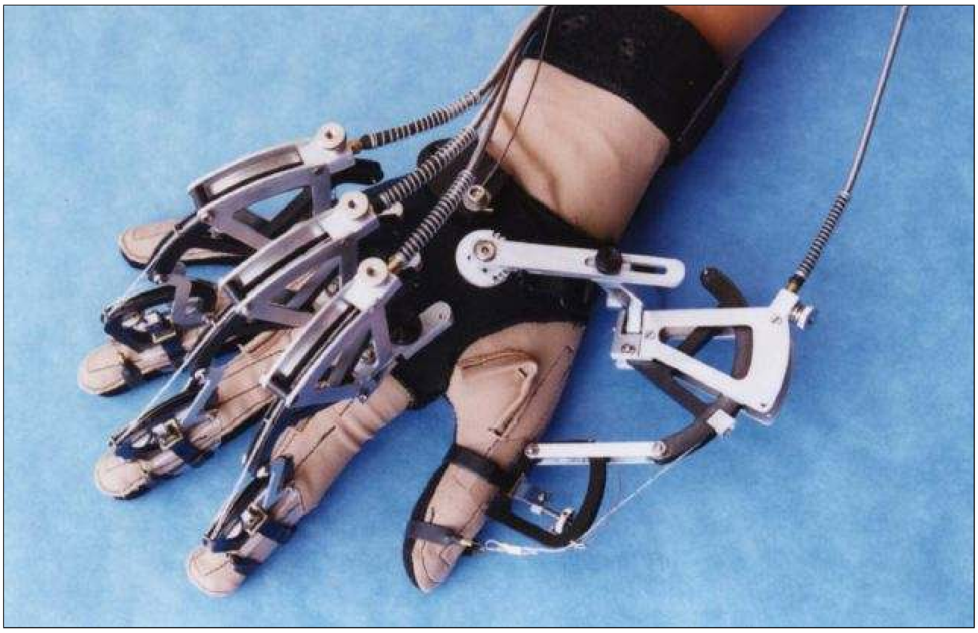


Figure 5.1.: **CyberGrasp glove**: With the CyberGrasp force feedback system, users are able to feel the size and shape of computer-generated 3D objects. Photo from Immersion Corporation site *http://www.immersion.com*

The CyberGrasp is a lightweight, force-reflecting exoskeleton that fits over a CyberGlove; and adds resistive force feedback to each finger. With the CyberGrasp force feedback system, users are able to feel the size and shape of computer-generated 3D objects in a simulated ”virtual world.”(see fig. 5.1). Grasp forces are produced by a network of tendons routed to the fingertips via the exoskeleton. There are five actuators, one for each finger, which can be individually programmed to prevent the user’s fingers from penetrating or crushing a virtual solid object. The high-bandwidth actuators are located in a small actuator module, which can be placed on the desktop. Originally developed under STTR contract to the United States Navy for use in telerobotic applications, the CyberGrasp system allows an operator to control a remotely-located robotic ”hand” and literally ”feel” the object being manipulated [DGT$^+$03].

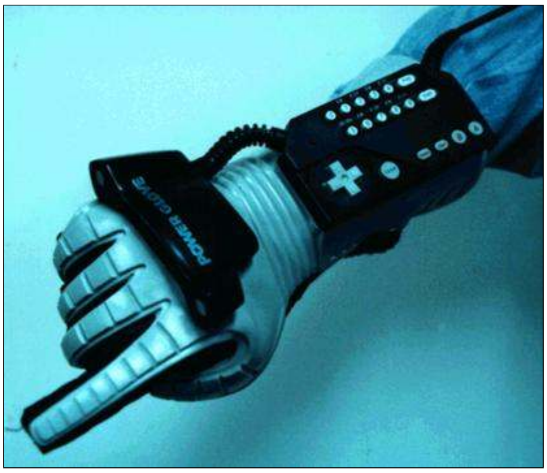


Figure 5.2.: **Mattel PowerGlove**: Measurement of finger flexure is performed by measuring the electrical resistance of a conductive ink which is painted onto the plastic structure that covers each finger. Photo from [Hol03].

The PowerGlove, developed by Mattel (the toy company) for use with the Nintendo game console, is a popular input device for low-end VR systems (figure 5.2). Position tracking is achieved by transmitting ultrasonic pulses from the two emitters on the back of the knuckles and measuring the time taken for these pulses to reach the 3 receivers mounted on the L-shaped bar. The time taken for the pulses to reach each receiver will be slightly different depending on the position of the glove in 3-D space. Measurement of finger flexure is performed by measuring the electrical resistance of a conductive ink which is painted onto the plastic structure that covers each finger. The small black box contains a microprocessor which co-ordinates emission and reception of pulses, as well as packaging the position and flexure data ready to be transmitted down the proprietary interface. When the user’s hand is moved it’s position is tracked by the glove. Usually a virtual hand is drawn on the screen to give the user visual feedback [GJC$^+$87].

The 5th Glove uses 5 fiber-optic flex sensors to generate finger-bend data for 5 fingers (see figure 5.3). It has an integrated pitch and roll sensor. The manufacturer claims the resolution is 256 po-

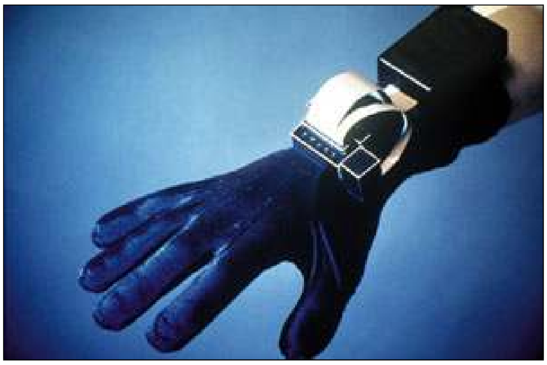


Figure 5.3.: **5DT 5th Glove**: The 5th Glove uses 5 fiber-optic flex sensors to generate finger-bend data for 5 fingers. It has an integrated pitch and roll sensor. Photo from Fifth Dimension Technologies site *http://www.5dt.com.*

sitions/finger. It can be connected to computer using RS-232 interface [Com96].

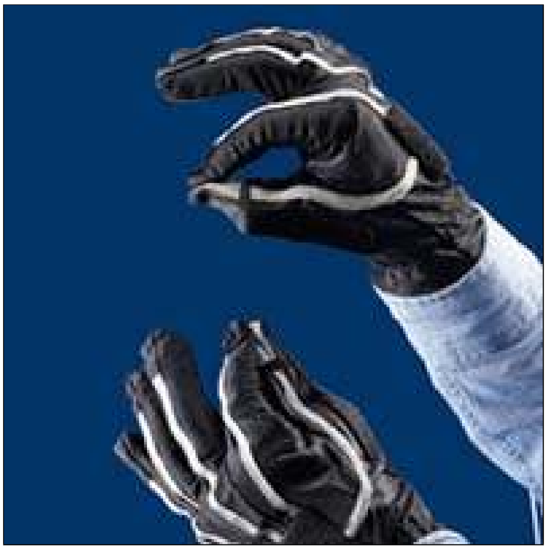


Figure 5.4.: **PINCH Glove**: The PINCH System uses cloth gloves with electrical sensors in each fingertip.

The PINCH glove system provides a reliable and low-cost method of recognizing natural gestures. Recognizable gestures have natural meaning to the user: a pinching gesture can be used to grab a virtual object, and a finger snap between the middle finger and thumb can be used to initiate an action. The fingers have a conductive element on the outside, and the system can recognize when fingers are touching. Allows users to ’pinch’ and ’grab’ virtual objects or initiate action. Each finger can be programmed to generate different actions.

Users can program as many functions as can be remembered. The PINCH System uses cloth gloves with electrical sensors in each fingertip. Contact between any two or more digits completes a conductive path, and a complex variety of ”pinch” gestures can be programmed into applications. A pinching gesture can be used to grab a virtual object and a finger snap can be used to initiate an action.

Active gloves

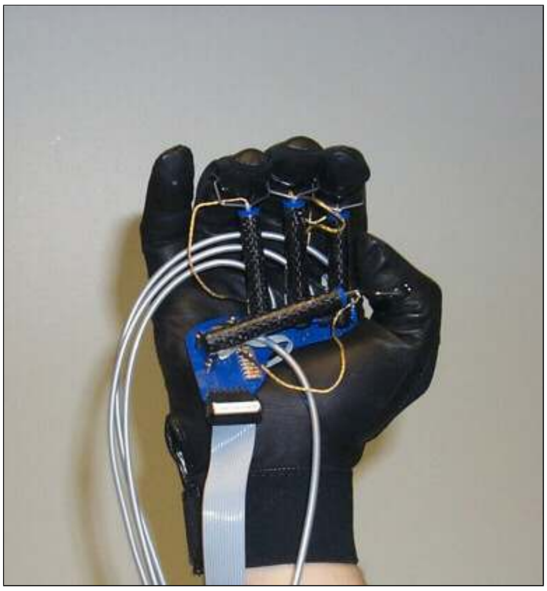


Figure 5.5.: **Rutgers Masters II**: RM-II reads hand gestures (hand-master) and displays forces (haptic-display) to four finger in real time. Photo from Rutger University site *http://www.caip.rutgers.edu*.

There are currently only two force reflecting gloves available: the Rutger Masters II, and the Exoskeleton. The Rutgers Masters II (RM-II), was realized at the Rutgers University in 1995 [DGN95], and it is a human-machine interface for haptic display in virtual reality and telerobotics. RM-II reads hand gestures (hand-master) and displays forces (haptic-display) to four finger in real time. The system has been integrated in several VR simulations. It uses pneumatic pistons which are mounted on the palm of the glove. When a virtual object is placed in the virtual hand, the user’s hand can close around it. When the fingers would meet resistance from the object in reality, the pressure in the pistons is increased, giving the sensation of resistance from the virtual object [DGN95].

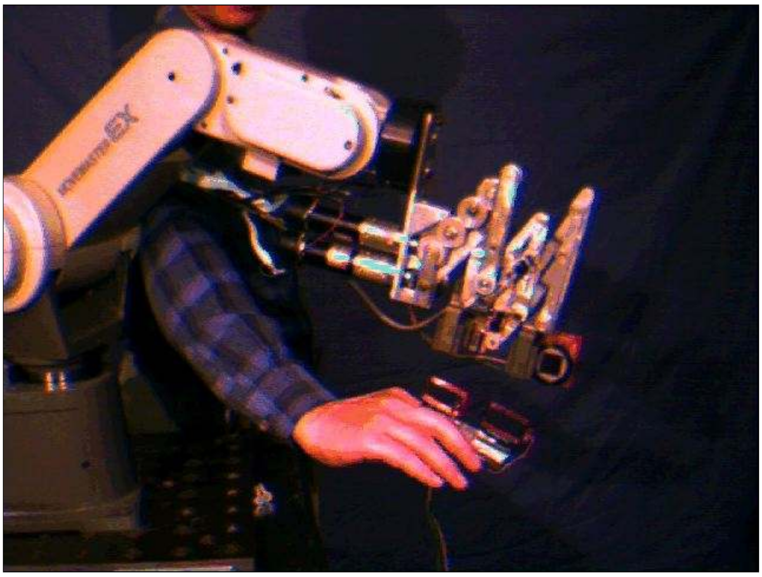


Figure 5.6.: **Iowa Exoskeleton**: based on theapplication of electromagnetic principles to couple the human hand with a robotic manipulator. Photo from Iowa University site *http://www.icemt.iastate.edu.*

The Exoskeleton, realized in Iowa State University, is based on the application of electromagnetic principles to couple the human hand with a robotic manipulator. Using this approach, the forces are transmitted between the robot exoskeleton and the human without using mechanical attachments to the robot. This system (see figure 5.6) provides the human user with an interface to the computer generated virtual world that will apply computer generated forces to the digits of the human operator's hand according to an arbitrary model residing within the computer program [LW94].

Suits

In the market, the only available passive virtual-reality body suit is manufactured by Aura. The Aura Interactor (see figure 5.7) works by ”listening” to any game system's audio output, triggering an Aura Magnetic actuator within the vest's core, which in turn creates body-pulsing vibrations keyed to the on-screen action. Controls allow the output to be fine-tuned from a feather touch to intense pounding, and can provide a way to suppress the music from some games if desired. The Interactor also works with standard TV, VCR, and music CDs, adding the sensation of feeling to these activities as well.

There are very few suits in the market (active or inactive). It is estimated that in the future, there will be more commercial suits like this

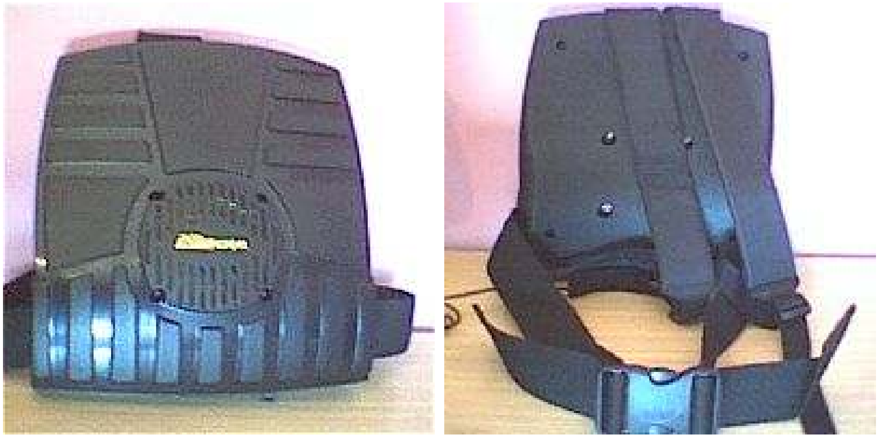


Figure 5.7.: **Aura Interactor**: it creates body-pulsing vibrations keyed to the on-screen action. Photo taken from Aura systems site.

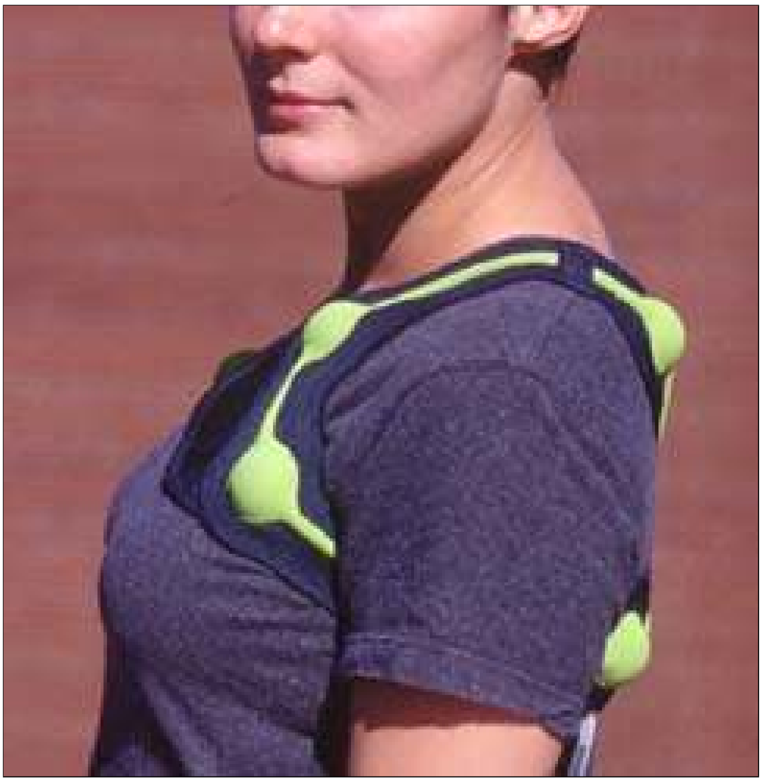


Figure 5.8.: **Tactile vest**: a wearable device that provides haptic feedback to the torso, based on Nitinol, a shape memory alloy (SMA). Photo taken from Berkeley University site *http://kingkong.me.berkeley.edu.*

in the market. Manufacturers have so far focused on force feedback for hands or arms, but there is a big potential for the use of body-based force reflecting suits. Relatively to this, recently remarkable efforts were aimed to the design of a robust, versatile actuator for a Tactile Vest, a wearable device that provides haptic feedback to the torso [MJ03] (see figure 5.8). The technology proposed is based on Nitinol, a shape memory alloy (SMA). Nitinol is capable of generating large stresses and strains, but is small and flexible and able to fit in the thin, dynamic workspace of a vest.

5.2.2. Ground-based devices

Hand controllers

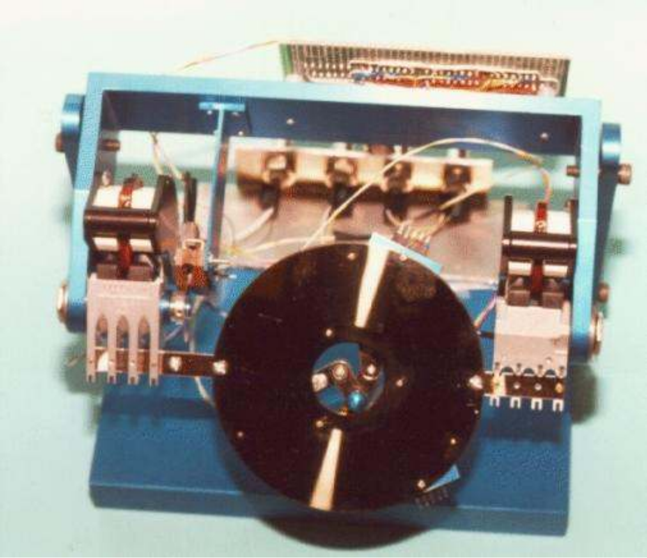


Figure 5.9.: **Pen-based Force Display**: operators interact with it using either the fingertip or a freely held pen-like tool. Photo taken from University of Washington web resource *http://brl.ee.washington.edu.*

In last few years, haptics community has made many efforts in developing new tipologies of hand controllers for providing force feedback. The principal ones are:

- The Pen Based Force Display
- The High Bandwidth Force Display
- The Excalibur Display
- The Carnegie Mellon University Magnetic Levitation Haptic Interface

- Sarcos Dextrous Master
- The University of Tsukuba Haptic Master
- SHade
- Delta Haptic Interface
- PHANToM Haptic interface

The Pen Based Force Display is a direct-drive, parallel, redundant, three DOF haptic device, designed to provide force feedback information generated by either a master-slave system or a virtual simulation. The operator interacts with it using either the fingertip or a freely held pen-like tool (see figure 5.9). The system is composed of a 2 DOF planar, parallel, redundant manipulator that can be connected to a more powerful actuator to provide a third DOF. The workspace is 150 or 200 mm^2. Spatial resolution is about 0.02 mm. Force output is 1-1.50 N. Static friction is 0.01 N (or 1% of output force). Its mass is 10g. It is controlled by a PC. The sampling rate is 1 kHz. [PB95]

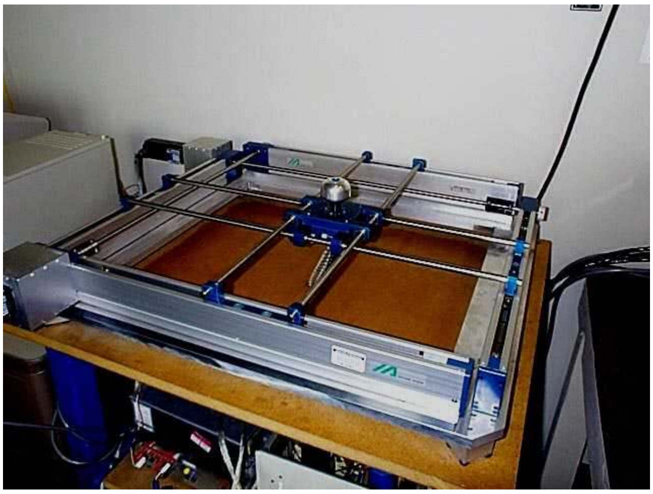


Figure 5.10.: **High Bandwidth Force Display**: It consists of a kinematically simple Cartesian mechanism driven by brushless DC motors through steel cable transmission. Photo taken from University of Washington web resource *http://brl.ee.washington.edu.*

The High Bandwidth Force Display is a two DOF force display for study of simulated interaction with heavy and stiff virtual objects in the plane. It consists of a kinematically simple Cartesian mechanism driven by brush less DC motors through steel cable transmission. The workspace is 300 mm x 400 mm (see figure 5.10). Spatial resolution is about 0.015 mm. Continuous force output is 400N and the peak is 1200 N. Static friction is 3.5 - 5.0 N (or 1kg. Device is controlled by a 486 PC. The sampling rate is 1 kHz. Virtual stiffness of up to 70 kN/m can be sensed by the device. [MB98]

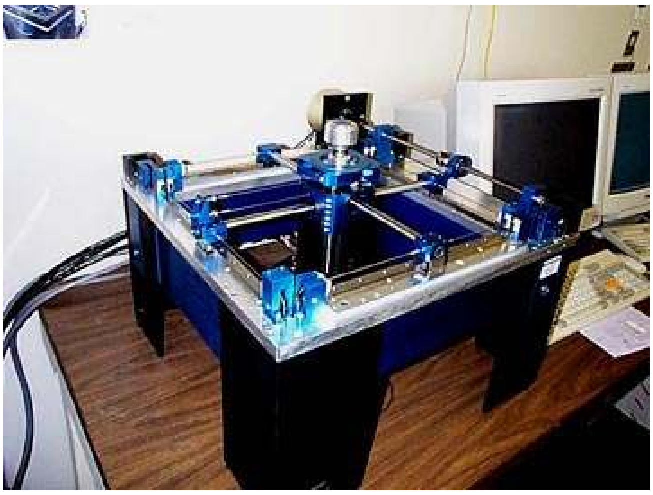


Figure 5.11.: **Linear Haptic Display - Excalibur**: is a 3 degree-of-freedom haptic device, built for maximum workspace, force output, and structural stiffness. Photo taken from University of Washington web resource *http://brl.ee.washington.edu.*

The Linear Haptic Display (LHD) - Excalibur is a 3 degree-of-freedom haptic device, built for maximum workspace, force output, and structural stiffness. Its hallmark is a patented steel cable transmission system which enables high forces and high rigidity in the three orthogonal axis of translational motion. The motors are mounted on the base of the mechanism, so that only light linkage elements move with the hand-grip. The workspace is 300x300x200 mm^3 (see figure 5.11). Spatial resolution is about 0.008 mm. Continuous force output is 100 N and the peak is 200 N. Static friction is 5.0N. Its mass is 3 kg. [AH99, Ada99]

The Carnegie Mellon University Magnetic Levitation Haptic Interface is based on a recently developed magnetic levitation technology. The magnetic levitation technology uses Lorentz forces to stably levitate and control a rigid body (which includes the handle through which the user interacts) in six degrees of freedom, giving a new and heretofore unexplored physical basis for haptic interaction. Robust realistic physical simulation, coupled with high-bandwidth six-degree-of-freedom force reflection, has the potential to greatly improve the state of the art in the feel, performance, and capabilities of virtual environment systems for use in a wide range of human activi-

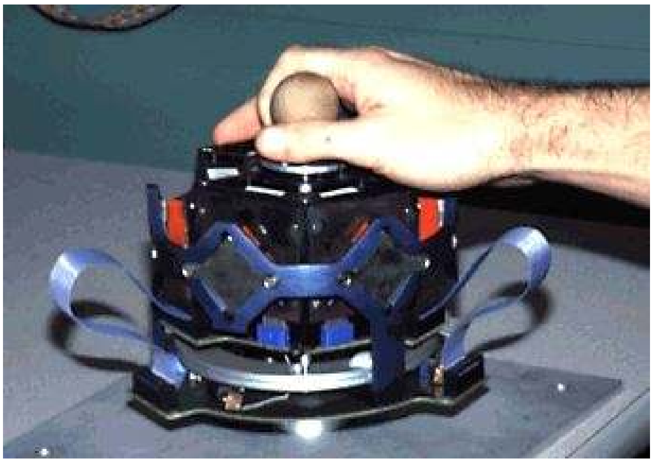


Figure 5.12.: **CMU Magnetic Levitation Haptic Interface**: the magnetic levitation technology uses Lorentz forces to stably levitate and control a rigid body. Photo taken from Carnegie Mellon University site *http://www-2.cs.cmu.edu/ msl/*.

ties [BHB99]. The main drawback of magnetically levitated devices is small workspace, and the main advantage is the superior precision (see figure 5.12).

The University of Tsukuba Haptic Master [Iwa90] is based on a six-degree-of-freedom manipulator that employs a parallel mechanism. Haptic master system employs three sets of parallelogram linkages(pantographs) instead of linear actuators. Each pantograph is driven by three DC motors, powered by PWM(Pulse Width Modulation) amplifiers.The top end of the pantograph is connected to a vertex of the top platform by a spherical joint (see fig. 5.13). The working space of the center of top platform is a spherical volume whose diameter is approximately 40 cm, much higher than magnetically levitated masters. Each joint angle of the manipulator is measured by potentiometers. The maximum payload of the manipulator is 2.5 kilograms,which is more than a typical hand. This master can represent arbitrary direction and strength force. With this compact hardware,the operator can feel the physical characteristics of three types of virtual objects: hard surface, elastic surface, and fluid flow velocity.

One of the most complex force-reflecting devices built today is the Sarcos Dextrous Arm Master designed by Sarcos, Inc., in conjunction with the University of Utah's Center for Engineering Design and the Naval Ocean Systems Center (NOSC). Although it is primarily ground-based, by having attachments points at the forearm and the upper arm of the user, it has the advantages of an exoskeleton, such as a

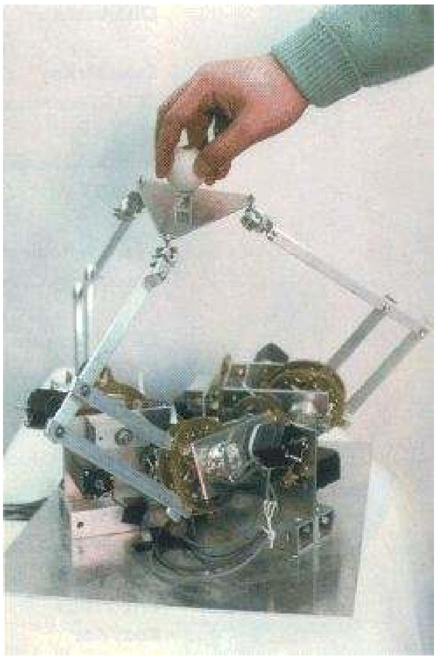


Figure 5.13.: **University of Tsukuba Haptic Master**: Haptic master system employs three sets of parallelogram linkages(pantographs) instead of linear actuators. Photo from web resource *http://intron.kz.tsukuba.ac.jp.*

large workspace comparable to that of the human arm (see fig. 5.14). This device utilizes high-performance hydraulic actuators to provide a wide dynamic range of force extension at relatively high bandwidth on a joint-by-joint basis for 10-DOF [NNHJ98].

Birglen and others [BGP+02] developed a 3-DOF haptic device, called SHaDe(Spherical Haptic Device), that allows a human operator to control motions while being subjected to force feedback. The mechanism presents the particularity of having only three degrees of freedom, leading to a simpler design and a more ergonomic utilization (see figure 5.15). Because of the spherical geometry, such a haptic device has several advantages, namely, a pure rotation around a point located inside the user's hand (no translations at this point), a large workspace, a comfortable usage, and precise manipulation with arm resting.

The Delta Haptic Device (see fig. 5.16) is a high performance haptic device based on the Delta manipulator and has 3 - 6 degrees of freedom: 3 translations from the parallel Delta structure and 3 rotations from a wrist module. Unlike other haptic mechanisms (which have either limited force capability or small workspace), the Delta Haptic Device is capable of providing large forces (up to 25N) over a large

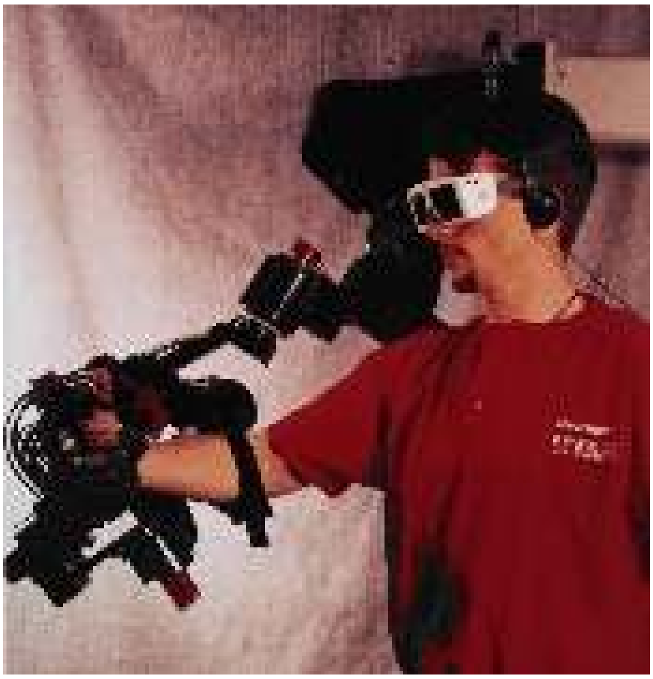


Figure 5.14.: **Sarcos Dextrous Arm Master**: this device utilizes high-performance hydraulic actuators to provide a wide dynamic range of force extension at relatively high bandwidth on a joint-by-joint basis for 10-DOF. Photo from Sarcos site *http://www.sarcos.com.*

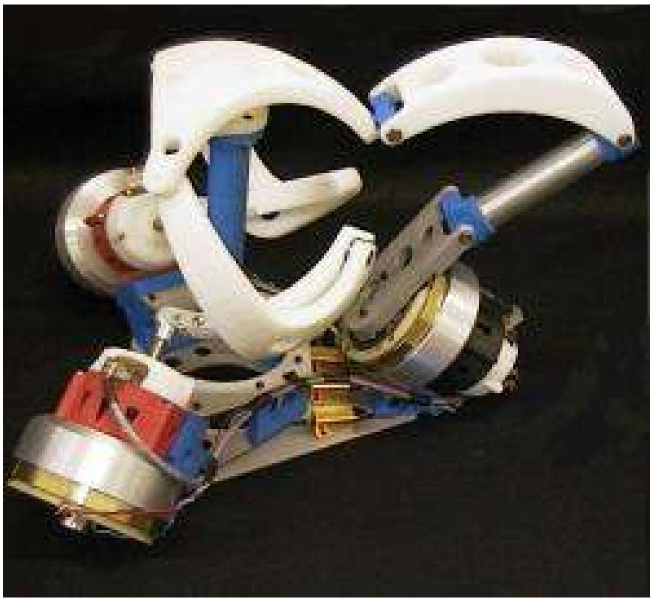


Figure 5.15.: **SHade**: Spherical HAptic DEvice. Photo from University of Laval site *http://www.robot.gmc.ulaval.ca/*.

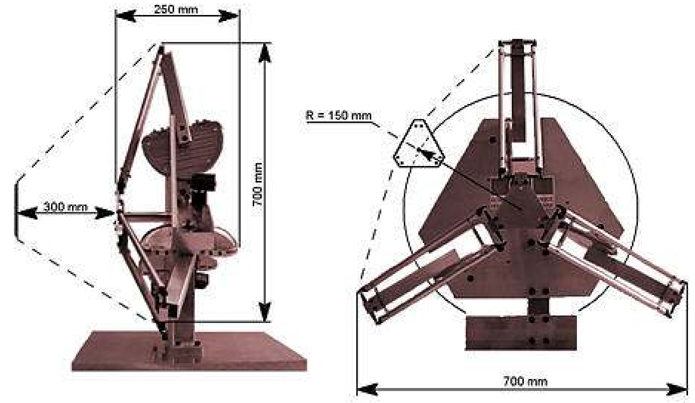


Figure 5.16.: **Delta Haptic Device**: it has 3 - 6 degrees of freedom: 3 translations from the parallel Delta structure and 3 rotations from a wrist module. Image from Stanford University web resource *http://robotics.stanford.edu.*

volume (30 cm diameter, 30 cm length).

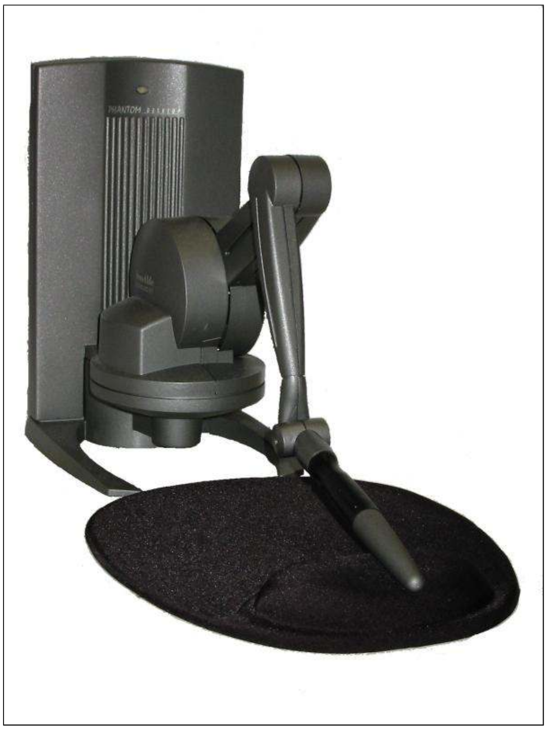


Figure 5.17.: **PHANToM Desktop**

The PHANTOM (Personal Haptic Interface Mechanism) haptic interface, is a device which measures a stylus end position and exerts a precisely controlled force vector on the stylus (see figure 5.17). In its simplest form, the PHANTOM can be thought of as a transmission between three DC brushed motors with encoders and the human finger. The device has enabled users to interact with and feel a wide variety of virtual objects and can also be used for control of remote manipulators [TK94].

5.3. PHANToM Desktop mechanical description

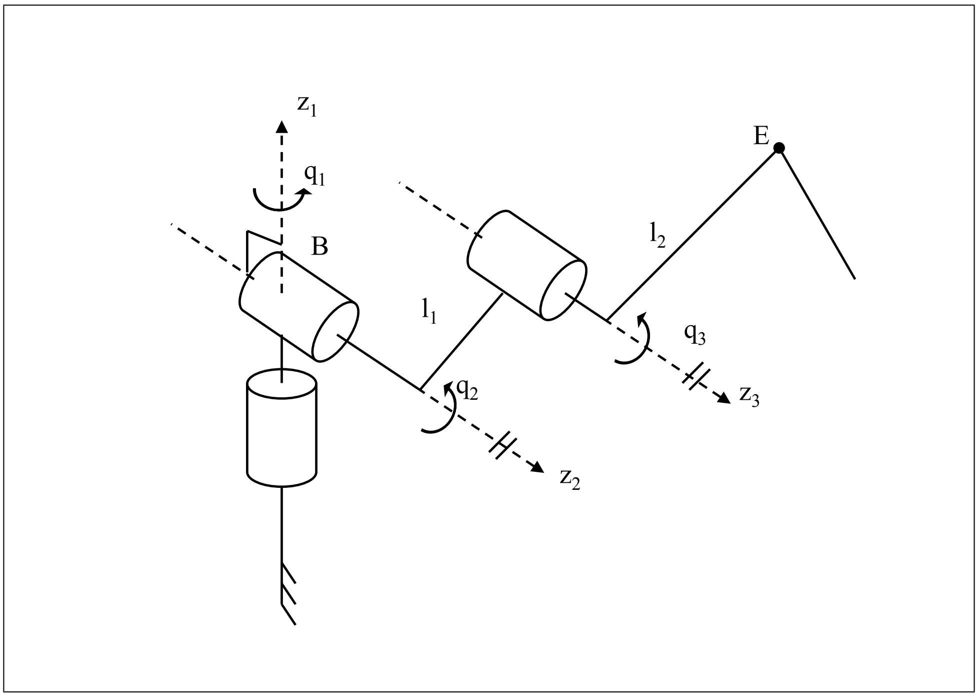


Figure 5.18.: **PHANToM Desktop conventional representation**: A PHANToM Desktop haptic interface conventional representation. The device can be considered as a serial 3R manipulator, with 3 rotational degrees of freedom.

We chose Sensable PHANTom haptic interfaces, for the following pratical reasons:

- the commercial availability and cost range
- the relatively small space requirement
- the pen-like handle, similar to surgical tools.

These features make the Sensable PHANToM suitable for the haptic simulation of surgical tasks. Figure 5.18 shows a conventional representation of the PHANToM models, that can be considered as a serial

3R manipulator, i.e. a spatial manipulator with three rotational degrees of freedom. The x, y and z coordinates of the users finger tip are tracked with the encoders, and the motors control the x, y and z forces exerted upon the user. Torques from the motors are transmitted through pre-tensioned cable reductions to a stiff, lightweight aluminum linkage. At the end of this linkage is a passive, three degrees of freedom gimbal attached to a thimble. Because the three passive rotational axes of the gimbal coincide at a point, there can be no torque about that point, only a pure force. This allows the users finger tip to assume any comfortable orientation. More importantly, because the user can be represented by a single point of friction-less sphere within the virtual environment, collisions and resulting interaction forces within the virtual environment are easily calculated. The PHANTOM has been designed so that the transformation matrix between motor rotations and endpoint translations is nearly diagonal. Decoupling the three motors produces desirable results in terms of back-drive friction and inertia [TK94]. In following subsections we provide the mechanical description of this haptic interfaces, in order to derive direct and inverse kinematics, as far as the equations of motion.

5.3.1. Direct kinematics

Figure 5.19 shows a schematic representation of PHANToM Desktop model, that we use to choose global and local reference systems, as far as lagragian parameters. In order to derive the haptic interface direct kinematic expressions, we need to attach a local frame to each joint in the mechanical system. Figure 5.20 shows the mechanical system, with global and local frames attached:

- B is the global base frame,
- $1, 2, 3, 4$ are the local frames attached to moving links
- 0 is the reset position global frame
- E is the end-effector local frame.

In figure 5.20 are also indicated the lagrangian parameters of the system, in Denavit-Harten-
berg notation (q_1, q_2, q_3) and in a modified notation $(\theta_1, \theta_2, \theta_3)$. We make our computations by using q_i coordinates, and then we will pass to θ_i in order to get simpler equations.

The frames attaching operation strictly follows the Denavit-Hartenberg notation, as described in Craig [Cra86]. The frames transforms are

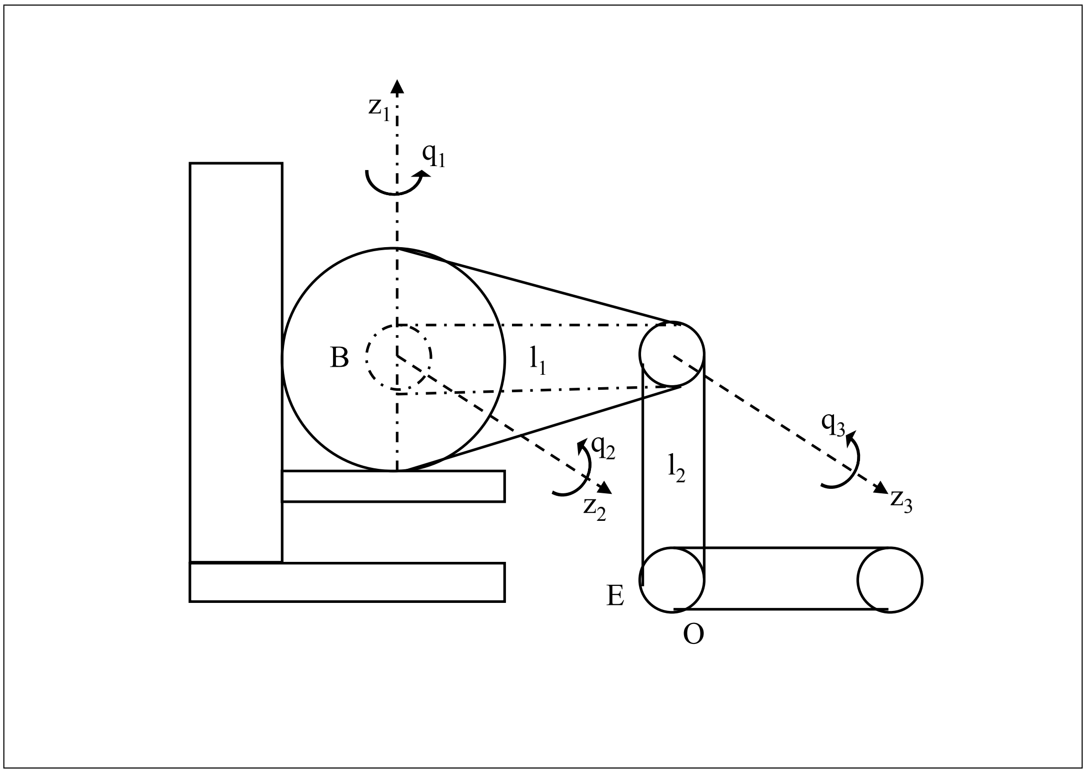


Figure 5.19.: **PHANToM Desktop scheme**: A PHANToM desktop schematic view. The mechanical system can be simplified by considering only two spatial links and three rotational joints.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$\frac{\pi}{2}$	0	0	θ_2
3	0	l_1	0	θ_3
4	0	l_2	0	0

Table 5.1.: Denavit-Hartenberg link parameters for PHANToM haptic interface.

built by using the parameters in table 5.3.1. According to that table, the PHANToM haptic interface can be considered as a serial manipulator with three rotational degrees of freedom. The affine transforms that link the various frames attached to the mechanical system are the following:

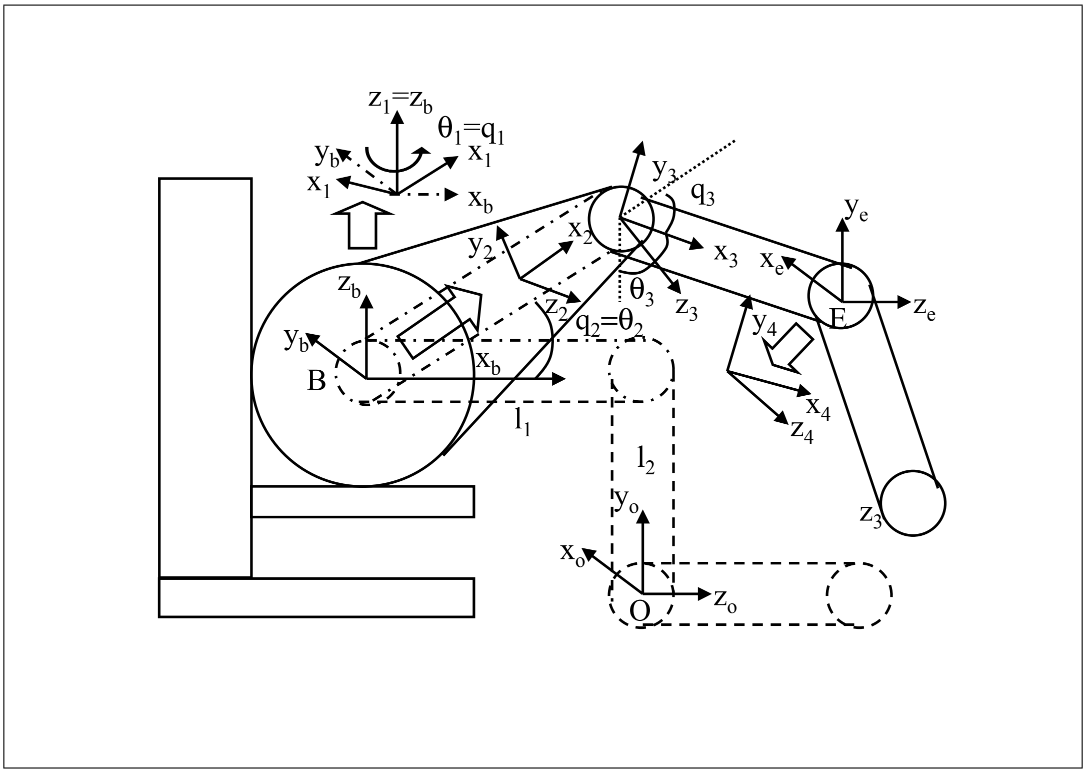


Figure 5.20.: **PHANToM Desktop frames**: A PHANToM desktop schematic view, with frames and joint motion indications.

$$
{}^{B}_{1}T := \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T := \begin{bmatrix} \cos q_1 & -\sin q_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.1)
$$

$$
{}^{2}_{3}T := \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & l_1 \\ \sin q_3 & \cos q_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}_{4}T := \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.2)
$$

$$
{}^{O}_{B}T := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ 1 & 0 & 0 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{4}_{E}T := \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.3)
$$

By composing the previous transform matrix, we get the direct kinematics equations, that relates the end-effector position to lagrangian parameters. If we apply the following lagrangian variables transforms:

$$\left\{ \begin{array}{l} \theta_1 = q_1 \\ \theta_2 = q_2 \\ \theta_3 = q_3 - q_2 - \frac{\pi}{2} \end{array} \right. \tag{5.4}$$

we are able to derive the position of the end–effector, as function of lagrangian parameters $(\theta_1, \theta_2, \theta_3)$. Hence, the complete transform matrix, that relates end–effector local frame to the reset position global frame, is:

$${}^{0}_{E}T := {}^{0}_{B}T \cdot {}^{B}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{E}T \tag{5.5}$$

and we obtain,

$${}^{0}_{E}T\left(\theta_1, \theta_2, \theta_3\right) := \left[\begin{array}{cccc} \cos\theta_1 & -\sin\theta_1\sin\theta_3 & \cos\theta_1\cos\theta_3 & \sin\theta_1\left(l_1\cos\theta_2 + l_2\sin\theta_3\right) \\ 0 & \cos\theta_3 & \sin\theta_3 & l_2 - l_2\cos\theta_3 + l_1\sin\theta_2 \\ -\sin\theta_1 & -\cos\theta_1\sin\theta_3 & \cos\theta_1\cos\theta_3 & -l_1 + \cos\theta_1\left(l_1\cos\theta_2 + l_2\sin\theta_3\right) \\ 0 & 0 & 0 & 1 \end{array} \right] \tag{5.6}$$

that is identical to the direct kinematics provided by Cavusoglu and others [CFT02]. The end – effector cartesian coordinates are hence

$$\left\{ \begin{array}{l} x_e = l_1\sin\theta_1\cos\theta_2 + l_2\sin\theta_1\theta_3 \\ y_e = l_2 - l_2\cos\theta_3 + l_1\sin\theta_2 \\ z_e = -l_1 + l_1\cos\theta_1\cos\theta_2 + l_2\cos\theta_1\sin\theta_3 \end{array} \right. \tag{5.7}$$

5.3.2. Inverse kinematics

The manipulator inverse kinematics is derived by simple view inspection. If we consider the figure 5.21, we have that the parameter θ_1 can be directly derived by

$$\tan\theta_1 = \frac{x_e}{z_e + l_1} = \frac{\sin\theta_1\left(l_1\cos\theta_2 + l_2\sin\theta_3\right)}{\cos\theta_1\left(l_1\cos\theta_2 + l_2\sin\theta_3\right)} \tag{5.8}$$

that leads to the θ_1 variable,

$$\theta_1 = \tan^{-1}\left(x, z + l_1\right). \tag{5.9}$$

If we consider lateral view in figure 5.22, we can derive R and r,

$$R = \sqrt{x_e^2 + \left(z_e + l_1\right)^2} \tag{5.10}$$

$$r = \sqrt{R^2 + \left(y_e - l_2\right)^2} \tag{5.11}$$

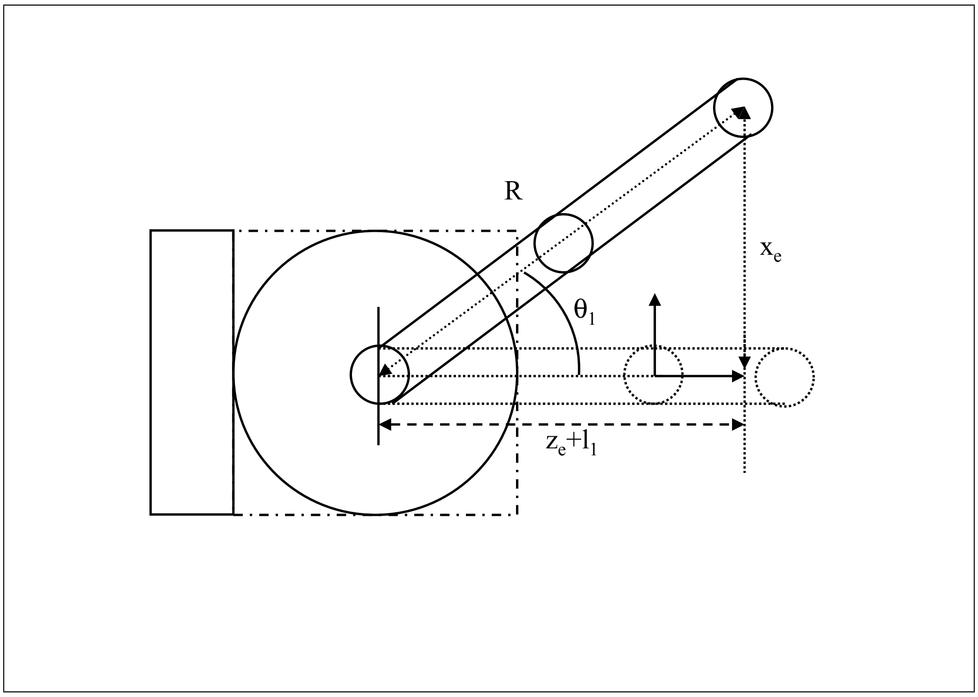


Figure 5.21.: **PHANToM schematic top view**: PHANToM schematic top view, used to derive inverse kinematics.

By view inspection, and by applying the cosines law, we derive

$$\beta = \tan^{-1}(y_e - l_2, R), \tag{5.12}$$

and

$$l_2^2 = l_1^2 + r^2 - 2l_1 r\cos\gamma \Rightarrow \gamma = \cos^{-1}\left(\frac{l_1^2+r^2-l_2^2}{2l_1 r}\right) \tag{5.13}$$

Since in the physical workspace we have $\gamma > 0$, hence $\theta_2 = \gamma + \beta$, so we can obtain

$$\theta_2 = \cos^{-1}\left(\frac{l_1^2+r^2-l_2^2}{2l_1 r}\right) + \tan^{-1}(y_e - l_2, R). \tag{5.14}$$

Applying cosines law again, we can write

$$r^2 = l_1^2 + l_2^2 - 2l_1 l_2\cos\alpha \Rightarrow \alpha = \cos^{-1}\left(\frac{l_1^2+l_2^2-r^2}{2l_1 l_2}\right), \tag{5.15}$$

and, in a physical workspace $\alpha > 0$, so $\theta_3 = \alpha - \frac{\pi}{2} + \theta_2$, then

$$\theta_3 = \cos^{-1}\left(\frac{l_1^2+l_2^2-r^2}{2l_1 l_2}\right) + \theta_2 - \frac{\pi}{2} \tag{5.16}$$

The previous relationships completely solve the inverse kinematics problem. The inverse kinematic equations provide joint parameters values, as functions of the end–effector cartesian coordinates.

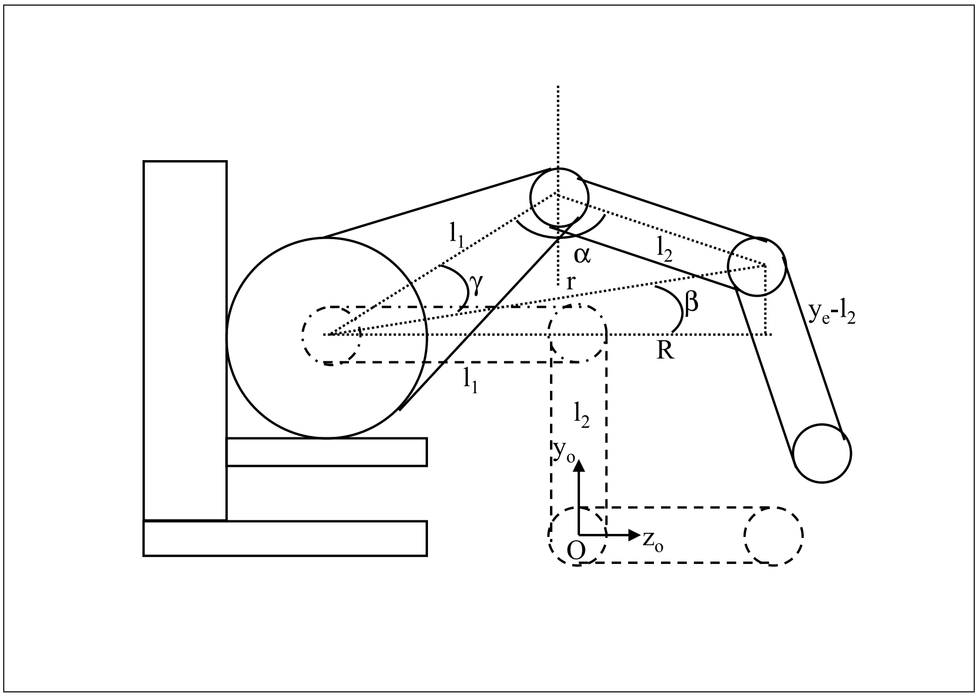


Figure 5.22.: **PHANToM schematic lateral view**: PHANToM schematic lateral view, used to derive inverse kinematics.

5.3.3. Motion equations

Relatively to the dynamic equations of motion for the manipulator, we need to take into account the forces applied to the end-effector E, that are applied by interface motor torques (τ_1, τ_2, τ_3). Joint frictions are, from a dynamical point of view, to be considered too, and weights, that act on links barycenters. A dynamics simplified model is obtained, by considering only two cylindric hollow rods, with masses m_1 and m_2, radii R, and thickness r, and lengths l_1 and l_2, las shown in kinematic equations. We derive motion equations, by using the Lagrange formulation.

Let T be the total mechanical system kinetic energy, V the system potential energy, and D the dissipated energy; the Lagrange equations for the mechanical system take the following form:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_i}\right) - \frac{\partial T}{\partial \theta_i} + \frac{\partial D}{\partial \dot{\theta}_i} + \frac{\partial V}{\partial \theta_i} = \tau_i, (i = 1, 2, 3) \tag{5.17}$$

To solve the derivation problem, we need now to express the system energies, as function of the lagrangian parameters $(\theta_1, \theta_2, \theta_3)$. First of all, we consider the kinetic contribution of the two links l_1 and l_2:

$$\begin{cases} T = T_1 + T_2 \\ T_1 = \frac{m_1 v_1^2}{2} + \frac{I_1 \dot{\theta}_1^2}{2} \\ T_2 = \frac{m_2 v_2^2}{2} + \frac{I_2 \dot{\theta}_2^2}{2} \end{cases} \tag{5.18}$$

where v_i is the link i center velocity, $\dot{\theta}_i$ is the link i angular velocity, I_i is the inertial moments related to the rotation axis. In the same manner, we are able to express the dissipation energy as

$$D = d\left(\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2\right) \tag{5.19}$$

where d is the rotational friction of the joints, that we assume the dame for each joint. Finally, the potential energy of the system is given by

$$V = g\left(m_1 h_1 + m_2 h_2\right) \tag{5.20}$$

where h_i is the y coordinate (height) of the link i center of mass. By expressing each single term as function of material and geometric properties of the mechanical system, we find that Lagrangian dynamic equations are in the form:

$$\begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{5.21}$$

where

$$\begin{cases} M_{11} = m_1\left[\delta^2\left(1+\sin^2\theta_2\right) + \frac{l_1^2}{3}\right] + m_2\left[\delta^2\left(1+\cos^2\theta_2\right) + \lambda^2\right] \\ M_{22} = m_1\left(\frac{l_1^2}{3} + \delta^2\right) + m_2\left(l_1^2 + \delta^2 + \frac{l_2^2}{12}\right) \\ M_{23} = -\frac{l_1 l_2 m_2 \sin(\theta_2-\theta_3)}{2} \\ M_{32} = M_{23} \\ M_{33} = m_2\left(\frac{l_2^2}{3} + \delta^2\right) \end{cases} \tag{5.22}$$

are the inertial contributions, with $\delta^2 = \frac{r^2-2rR+2R^2}{4}$ and $\lambda^2 = l_1^2\cos^2\theta_2 + l_1 l_2\cos\theta_2\sin\theta_3 + \frac{l_2^2}{3}\sin^2\theta_3$

$$\left\{ \begin{array}{l} C_{11} = 2d + \left[-\frac{l_1 l_2 m_2}{2} \sin\theta_2 \sin\theta_3 + \frac{\sin 2\theta_2}{6} \left(3m_1\delta^2 - m_1 l_2^2 - 3m_2 l_1^2\right) \right] \dot{\theta}_2 \\ + \left[\frac{l_1 l_2 m_2}{2} \cos\theta_2 \cos\theta_3 + \frac{\sin 2\theta_3}{6} \left(m_2 l_2^2 - 3\delta^2\right) \right] \dot{\theta}_3 \\ C_{12} = \left[-\frac{l_1 l_2 m_2}{2} \sin\theta_2 \sin\theta_3 + \frac{\sin 2\theta_2}{6} \left(3m_1\delta^2 - m_1 l_2^2 - 3m_2 l_1^2\right) \right] \dot{\theta}_2 \\ C_{13} = \left[\frac{l_1 l_2 m_2}{2} \cos\theta_2 \cos\theta_3 + \frac{\sin 2\theta_3}{6} \left(m_2 l_2^2 - 3\delta^2\right) \right] \dot{\theta}_3 \\ C_{21} = -C_{12} \\ C_{22} = 2d \\ C_{23} = \frac{l_1 l_2 m_2}{2} \cos\left(\theta_2 - \theta_3\right) \dot{\theta}_3 \\ C_{31} = -C_{13} \\ C_{32} = \frac{l_1 l_2 m_2}{2} \cos\left(\theta_2 - \theta_3\right) \dot{\theta}_2 \\ C_{33} = 2d \end{array} \right. \tag{5.23}$$

are the Coriolis contributions, and

$$\left\{ \begin{array}{l} N_2 = \frac{g l_1}{2} \left(m_1 + 2m_2\right) \cos\theta_2 \\ N_3 = \frac{g l_2 m_2}{2} \sin\theta_3 \end{array} \right. \tag{5.24}$$

are the gravitational contributions. These coefficients were obtained by using a Mathematica notebook. Readers interested on these scripts may refer to appendix A.

5.3.4. Validation

In order to validate the correctness of kinematics and dynamics equations previously derived, the following tests are performed:

- **Kinematics test:** results of direct kinematics equations are checked by applying inverse kinematics equations over a sampled space;
- **Dynamics test:** a model of the haptic interface is built with a Simulation Software called WorkingModel3D, and simulations are compared with results obtained by inverting the equations of motion.

Specifically, we make the **kinematics test** in the Matlab environment, by performing the following steps:

1. the cartesian space is sampled, in order to take into account most of the positions that can be reached by the manipulator;
2. for each position considered, the inverse kinematics is applied , in order to derive the joint parameters;
3. the loop is closed, by applying the direct kinematics to joint values previously computed, in order to get the cartesian position;

4. finally, the input cartesian point is compared with the output point: if the distance is less than a given threshold (i.e. the points are identical), the test is passed, otherwise there are problems in kinematics equations.

We applied this kinematics test procedure for a manipulator with the geometry and material parameters in table 5.3.4.

l_1	0.16 m
l_2	0.16 m
r	0.01 m
R	0.01 m
m_1	0.063 Kg
m_2	0.063 Kg
d	0 Kg $\frac{m^2}{s}$

Table 5.2.: Manipulator parameters used for kinematics and dynamics test.

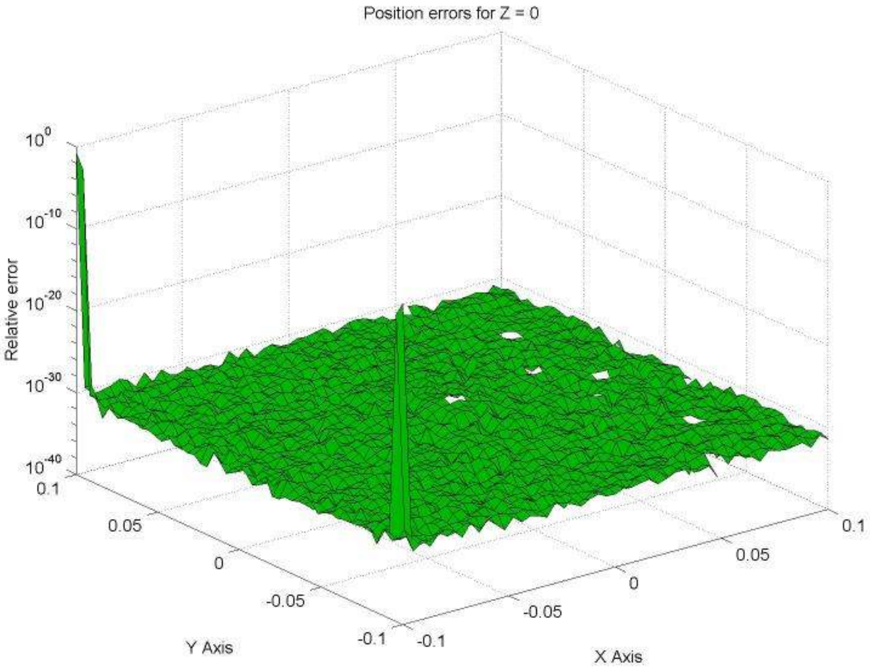


Figure 5.23.: **PHANToM kinematics test**: shows the relative error position, obtained by comparing input cartesian positions with those obtained by applying inverse and direct kinematics consecutively.

The figure 5.23 shows the the relative error position, obtained by comparing input cartesian positions with those obtained by applying inverse and direct kinematics consecutively. The space considered is a cube with length 0.2 cm, centered in the reset position, and the figure shows error values for the plane $Z = 0$.

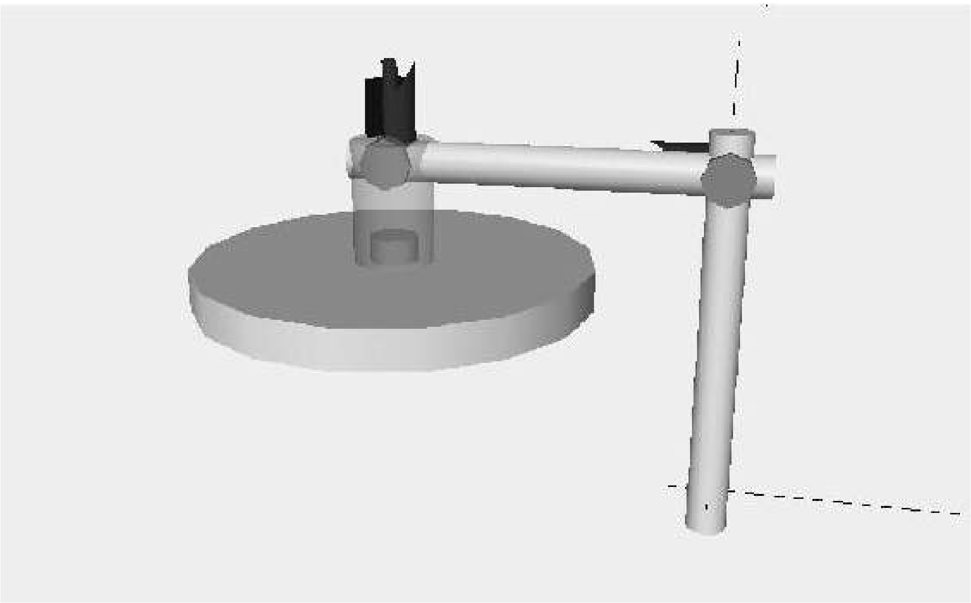


Figure 5.24.: **WorkingModel3d PHANToM model**: a simplified PHANToM model built inside the WorkingModel3d environment.

The **dynamics test** is more complicated and involves the usage of a mechanical simulation software called WorkingModel3D. The results obtained by the model built inside that software are compared with numerical results of the motion equations computed inside the Matlab environment. Specifically, the model built in WM3D is represented in figure 5.24, and contains the same material and geometry parameters of table 5.3.4. In this way, inverse dynamics simulations may be carried out, by applying some known motor torques, and by recording the joint values and the end-effector positions. These data are hence used as reference values to check the dynamic equations inside the Matlab environment. Actually, the dynamic equations 5.21 may be written in the vectorial form

$$M(\Theta)\ddot{\Theta} + C\left(\Theta,\dot{\Theta}\right)\dot{\Theta} + N(\Theta) = \mathrm{T}, \tag{5.25}$$

where Θ is the joint parameters vector, and T is the torque vector. If we introduce the vector $\Omega = \left[\theta, \dot{\theta}\right]^T$, we can reduce the dynamics equations to the ODE form $\dot{\Omega} = F(\Omega, t)$, i.e.

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} \omega_2 \\ M^{-1}(\omega_1)(\mathrm{T} - C(\omega_1, \omega_2)\omega_2 + N(\omega_1)) \end{bmatrix}. \tag{5.26}$$

The previous equation can be solved in Matlab environment, by an ODE solver like *ode115s*, in order to derive joint values for given torques. The results of numerical integration of the dynamic equations expressed in form 5.26 are then compared with the WM simulations. If numerical errors are under a given threshold, the correctness of motion equations is proved.

Figure 5.25 shows the results of an inverse dynamics simulation, performed by applying the torques $\mathrm{T} = [0.01Nm, 0.14Nm, 0.008Nm]$, with null initial joint positions and velocities, and the differences between the numerical solution of the inverse dynamics equations and the outputs from WorkingModel simulation. Results clearly show the correctness of the dynamic equations derived so far. Appendices A and B contain the Mathematica and MatLab scripts used to derive and test PHANToM kinematics and dynamics.

5.4. Conclusion

This chapter introduced the problems related to the usage of haptic interfaces for surgical simulation and training. Starting from a review of the haptic devices commonly used in virtual environments (section 5.2), we provided a mechanical description of Sensable PHANToM haptic interfaces (section 5.3), that are currently employed in our surgical training system. We chose Sensable PHANTom haptic interfaces, for the following pratical reasons:

- the commercial availability and cost range
- the relatively small space requirement
- the pen-like handle, similar to surgical tools.

These features make the Sensable PHANToM suitable for the haptic simulation of surgical tasks.

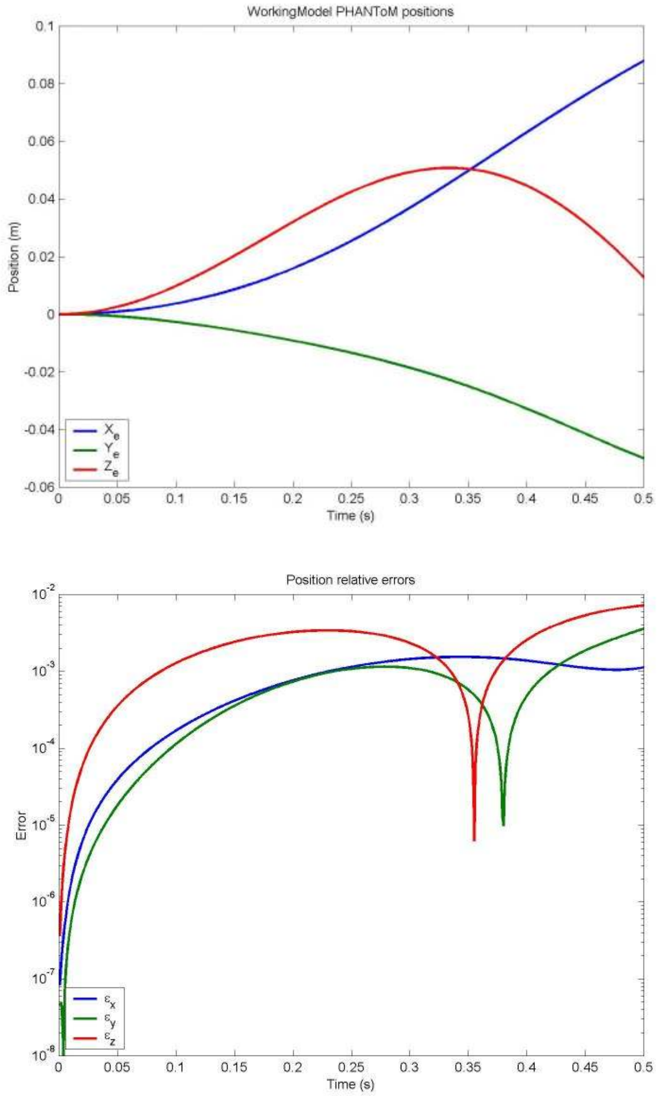


Figure 5.25.: **PHANToM dynamics test:** the figure shows the end-effector position during by an inverse dynamics simulation with given joint torques (left), and the difference between the numerical solution and the correspondent WM3D output simulation.

6. Real–Time Haptic Simulation

Bone dissection is the most important component of the surgical procedures that we consider in this thesis. In this chapter, burring and milling processes are presented and reviewed in order to individuate the principal influencing parameters, and the best way to reproduce the burring operation on haptic interfaces. Next, we discuss the haptic implementation of a bone cutting burr, that it is being developed as a component of our training system for temporal bone surgery. We use a physically motivated model to describe the burr–bone interaction, that includes haptic forces evaluation, the bone erosion process and the resulting debris.

6.1. Introduction

Haptic Virtual Environments are complex, multi-sensory simulators, aiming at enhancing the virtual experience by adding a feeling of touch to the visual interaction. Several recent studies point out the advantages of this augmented sensorial interaction [MBA02, AJMSP01]. With haptic systems, computational challenges become considerably more demanding, as the realistic experience has to result from the collaboration of two processes: one coordinating the visual system and the other tracking the position and updating the forces on the haptic device. Typically, haptic devices need to be updated at rates in the range of 300 to over 1000 Hz, otherwise they might provide degraded mechanical experience, such as bumpy motion or unpleasant vibrations. Hence, the need to provide real–time feedback to training surgeons, while simulating burring and related secondary effects, imposes stringent performance constraints. Our solution is based on a volumetric representation of the scene, and it harnesses the locality of the physical system evolution to model the system as a collection of loosely coupled components running in parallel on a multi-processor PC platform, as described in chapter 11. In our model, the burr bit is represented as a region of space that samples the volumetric bone data to construct the elastic reaction and friction forces that the bone opposes to the burring. The sampling algorithm is similar in spirit to the Voxmap PointShell approach [RPP$^+$01, MPT99a], even though here we use a volumetric region around the burr to select the bone voxels relevant to force calculation. Our algorithm for computing forces, loosely patterned on contact theory [LL86], is robust and a smooth function of the burr position. The computed forces are transfered to the haptic device via a *sample–estimate–hold* [ESJ97] interface to stabilize the system. Bone erosion is modeled by postulating an energy balance between the mechanical work performed by the burr motor and the energy needed to cut the bone, that it is assumed to be proportional to the bone mass removed. The hypothesis of friction proportional to erosion rate is called **Reye Hypothesis**, and it can be applied to all cases with presence of dry friction between contact surfaces [JP92]. The actual bone erosion is implemented by decreasing the density of the voxels that are in contact with the burr in a manner that is consistent with the predicted local mass flows. The rest of the chapter is structured as follows. Section 6.2 reviews the state of the art of physically based models used for haptic rendering, while section 6.3 deals with the models commonly employed in mechanical engineering to study milling and drilling processes. Section 6.4 is dedicated to the description of our bone–burr interaction model. Results are reported in section 6.5. Next chapter will describe an adaptive version of the same model, designed and developed in order to meet timing constraints, specially when large burr tips are employed.

6.2. Related work

The realism of the interaction with virtual objects depends on the type of models used to describe them. Non-physical models are simpler representations and therefore easier to update in real-time, but provide less realistic simulations. Physically based models are more complex representations, allowing for simulations of increased realism, but generally involve more computation. The border between the two categories is not sharp, e.g. mass-spring models are still relatively simple physically based models. At the top end of physically based models stand Finite Element (FE) models: they provide the most accurate simulations but also involve the highest computational costs [GM00]. Given a discrete representation of an elastic object, such as a finite set of nodes or vertexes linked in a tetrahedral mesh, one can fix a subset of these nodes in order to eliminate the rigid modes. Using the model of small displacements (linear elasticity), the equation of motion of the free nodes under the action of a field of external forces can be obtained by minimizing the total energy of the body. For details of this construction, see [BN98]. Linear elasticity is suitable for modeling small displacements, but may not be appropriate for handling large displacements [YJ00]. Over the last few years, haptic rendering of geometric models has received much attention. Most previous work addresses issues related to rendering the interaction between a probe point and 3D objects [RKK97]. This problem is characterized by high spatial coherence, and its computation can be localized. By contrast, we attack the problem of force rendering for arbitrary 3D polyhedral object-object interaction, which involves a substantially higher computational complexity. Force rendering of object-object interaction also makes it much more challenging to correctly cache results from previous computations. McNeely et al. [MPT99b] proposed point-voxel sampling, a discretized approximation technique for contact queries that generates points on moving objects and voxels on static geometry. This approximation algorithm is the first to offer run-time performance independent of the environment's input size by sampling the object geometry at a resolution that the given processor can handle. A recent approach proposed by Gregory et al. [GME$^+$00] is limited to haptic display of object-object interaction for relatively simple models that can be easily represented as unions of convex pieces. Kim et al. [KOLM02] attempt to increase the stability of the force feedback using contact clustering, but their algorithm for contact queries suffers from the same computational complexity. The idea of using multi-resolution representations for haptic rendering has been recently investigated by several researchers. Pai and Reissel [PR97] investigated the use of multi-resolution image curves for 2D haptic interaction. El-Sana and Varsheny [ESV00] proposed the construction of a multi-resolution hierarchy of the model during preprocessing. At run-time, a high-detail representation is used for regions around the probe pointer and a coarser representation farther away. The proposed approach only applies to haptic rendering using a point probe exploring a 3D model.

6.3. Milling and sinking models

A detail analysis of contact and cutting actions during the milling process presents remarkable difficulties that are shortly described in the sequel:

- the dynamic character of the forces that are originated by the contact between the cutting tool and the material;
- the presence of thermal effects due to the mechanical energy dissipated by friction;

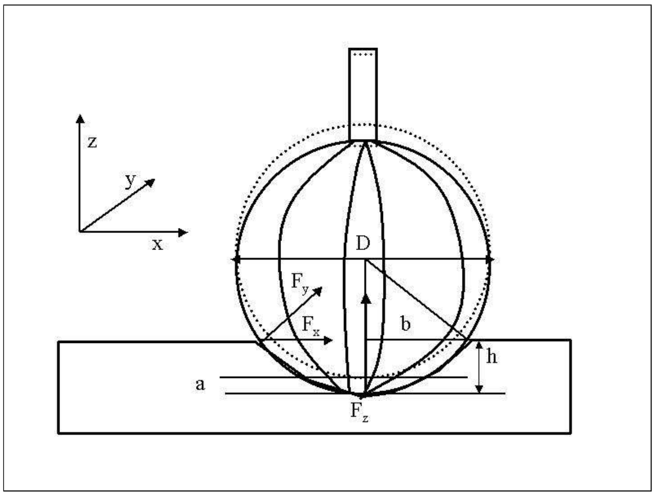


Figure 6.1.: **Sinking motion**: forces are generated by normal surface reaction, hole borders, and principal blades cutting action.

- the presence of plastic deformations that actually reach the material breakup
- the difficulty in accurately evaluating friction forces.

The problem is made even more complicated by the need to a–priori estimate the parameters involved in the cutting process. The choice made to overcome this problem, in common engineering problems, is to use simplified models of the milling process, that are derived from the classical theories of drilling and milling[Oxf55]. Within the frame of a simplified model some particular features of the milling process[CO57, JLG92] can be taken into account, such as the variation of the burr-bone contact surface area, and the cutting power limits imposed by the particular tool. In the sequel of this section, first we will discuss the power limits of the cutting tool, and which physical parameters have to be considered. Next, the principal type of cutting motions will be analyzed, namely the sinking motion corresponding to bone imprinting along the axis of the tool. We will also explain what are the difficulties related to the application of empirical models to the haptic display of surgical tasks.

6.3.1. Sinking motion modeling

Torque analysis

Many influencing factors act on the axial drilling force. For the analysis and the performance of experiments, this number of factors needs to be reduced. Some of them, though, cannot be neglected. In selecting the factors, the previous research and acquired experience need to be considered. In the majority of works that deal with the problems of bone drilling no mathematical model between the axial force and the influencing factors has been established, but rather only the significance of single factors has been determined.

We start by assuming that the forces applied to the burr bit, are originated by:

- principal blades cutting action
- central compression and surface normal reaction
- surface hole borders friction and scraping action
- dorsal friction resistance, that is particularly significant when blades are weared.

The resultant cutting force, can be decomposed along three principal directions, where z is the axis and sinking direction, x is, for each blade, tangent to the principal direction and normal to the surface, and y is normal to x and z, tangential to contact surface and normal to principal blade direction (see figure 6.1. For each blade, the force in y direction provides the principal contribution to cutting torque C, and can be used to compute so we can write

$$C \cong F_y \hat{b} \tag{6.1}$$

where $\hat{b}$ is the average torque arm, that can be computed, in the case of a spherical tool, with the following integral:

$$\hat{b} = \frac{1}{R} \int_0^R \sqrt{2Rh - h^2} dh = R \int_0^1 \sqrt{2x - x^2} dx = R \frac{\pi}{4}. \tag{6.2}$$

Now, we can employ the classical lamination theory, that evaluates the cutting force as $F_y = k_s A_z$, where k_s is the specific pressure, that decreases as chip thickness increases, and normally is substituted with the material break–down stress, and A_z is is the chip section area, that, in the worst case, can be approximated with aD, where a is the tool feed, and D is the bit diameter. Employing this assumption, the cutting torque can be expressed as

$$C \cong k_s A_z \pi \frac{R}{4} \cong \pi k_s \frac{aR^2}{2}. \tag{6.3}$$

The previous relationship can be used to evaluate the maximum feed that can be achieved, considering that the system power is limited. In fact, if we impose that $W \leq W_{max}$, and consider that $W \cong C\omega$, where ω is the burr angular velocity, we have

$$a \leq \frac{2W_{max}}{\pi k_s R^2 \omega}. \tag{6.4}$$

The previous relationship indicates that the feed is limited by the available power, by burr bit diameter, and by angular velocity. The equation (6.4) may be used to derive the working limits of the surgical tool during operation.

W_{max}	100 W
k_s	120 $\frac{N}{mm^2}$
R	3 mm
ω	50 $\frac{Krad}{sec}$

Table 6.1.: Typical surgical burr parameters.

Hence, considering the typical surgical burr parameters in table 6.3.1, we obtain that maximum sinking feed is about $0.6\frac{\mu m}{rev}$. This value will be used in next subsection, to derive the maximum reaction force that can be obtained during sinking motions.

Reaction force

A method commonly used to find the reaction force was derived by Shaw and Oxford [Oxf55]. Their approach is based upon a dimensional analysis of the drilling process, and expresses the reaction force F_a as function of the following process parameters: the burr radius R, the feed a, the ratio $q = \frac{c}{2R}$ between the central edge length c and the diameter $2R$, and the specific cutting energy u:

$$\frac{F_a}{8R^3 u} = k_1 \frac{a}{2R} f(q). \tag{6.5}$$

Assuming that the burr has a spherical bit, the q ratio can be neglected, and the specific cutting energy can be expressed as:

$$u \cong \frac{HB}{(2aR)^{0.2}} \tag{6.6}$$

where HB is the material Brinell hardness.

With these assumptions, we obtain a relationship for the reaction force

$$F_a \cong 10.97 HB \left(R^{0.8} a^{0.8}\right). \tag{6.7}$$

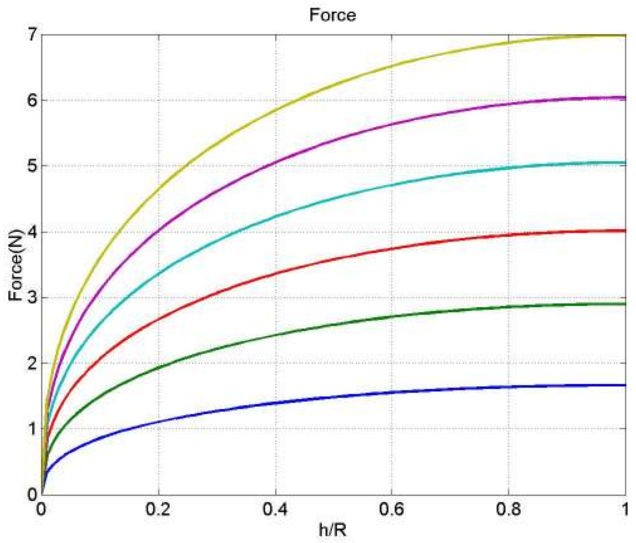


Figure 6.2.: **Force vs Depth**: the plots represent the force as function of ratio $\frac{h}{R}$, for various feeds from $0.1\frac{\mu m}{rev}$ to $0.6\frac{\mu m}{rev}$.

Now, a force model related to a burr bit with spherical shape must take into account the variation of the radius of the tool impression b , that depends on the ratio $\frac{h}{R}$ between the sinking depth h and the tool radius R. By making some simple geometrical considerations about the spherical shape of the tool (see figure 6.1), we can derive:

$$b = R \cdot \sqrt{\frac{h}{R}\left(2 - \frac{h}{R}\right)}. \tag{6.8}$$

By substituting R in equation 6.7 with b, we are able to write out a simplified expression for the reaction force acting on a spherical tool during the drilling process:

$$F = 10.97HB \cdot a^{0.8} \cdot R^{0.8} \cdot \left[\frac{h}{R}\left(2 - \frac{h}{R}\right)\right]^{0.4}. \tag{6.9}$$

Figure 6.2 shows the force plots obtained using equation 6.9, when considering a as a parameter and the relative impression $\frac{h}{R}$ as a variable. We assumed a HB hardness value for temporal bone of 50. Six different values of a (from $0.1\frac{\mu m}{rev}$ to $0.6\frac{\mu m}{rev}$) are considered. In previous subsection, we obtained that, for a typical surgical burr, maximum feed value is $0.6\frac{\mu m}{rev}$. So, according to equation 6.9, the maximum normal force that we can expect in a surgical operation is $7N$. During normal burr movements, force modulus is under $5N$, so Sensable PHANToM haptic interfaces, that provides $5N$ forces, are suitable for the rendering of bone–burr contact.

Despite its simplicity, the sinking model so far described has some drawbacks, related to the fact that the involved parameters cannot be accurately evaluated. Moreover, the model is only valid for a working burr, since it does not take into account the rigid contact between burr bit and material. Therefore, it cannot work in cases of pure elastic contact, occurring when the burr does not rotate and the blades do not exert their cutting action. This is a frequent situation during a surgical operation. Hence, to circumvent these problems, we derived an alternative model, described in the following section.

6.4. Bone–burr interaction model

A detailed mechanical description of a rotating burr cutting bone is complicated because it involves tracking the continuously changing free surface of the material being cut; the impact of the burr blades on the surface; the resulting stress distribution in the material; and the consequent plastic deformation and break–up. Sinking and drilling models reported in previous section are not able to take into account the elastic impact of the blades with the free surface. Hence, to circumvent these complications, we have divided the cutting process in two successive steps. The first estimates the bone material deformation and the resulting elastic forces, given the relative position of the burr with respect to the bone. The second estimates the local rate of cutting of the bone by using a – postulated – energy balance between the mechanical work performed by the burr motor and the energy needed to cut the bone, that it is assumed to be proportional to the bone mass removed.

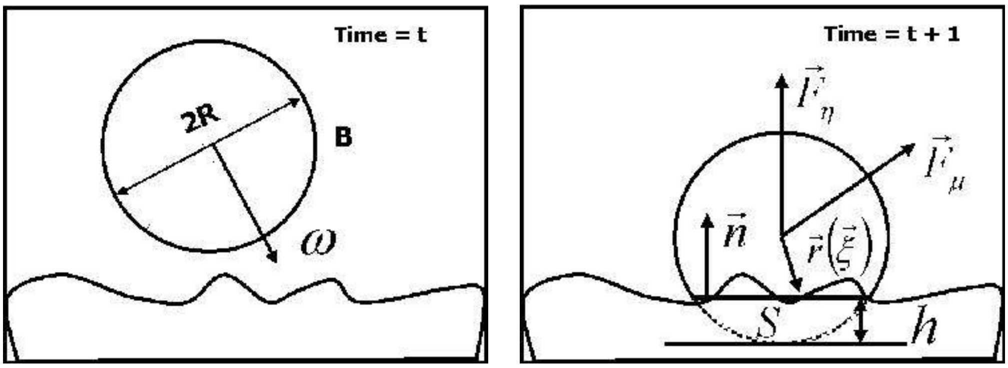
We will first describe this approach on a continuum model and then specialize the results to a discretized voxel grid.

6.4.1. Continuum description

Force evaluation

In figure 6.3 we represent two successive instants, at time t and $t+1$, of an idealized version of a surgeon burr. The burr has a spherical bit, of radius R, that is rotating with angular velocity $\vec{\omega}$. At time step t the burr is just outside the bone material, while at the next time step it is intersecting the bone surface. In the following, we will refer to the sphere representing the burr bit as B, and to the "contact surface" between the burr and the bone as S.

All the relevant geometrical information is contained in the volumetric distribution of the bone material. We use a characteristic



(a) Time=t

(b) Time=t+1

Figure 6.3.: **The impact of burr on bone**. Here we represent two successive instants, at time t and $t+1$, of an idealized version of a surgeon burr.

function $\chi(\vec{r})$ to indicate the presence/absence of bone, where $\vec{r}$ is measured from the center of B. The first two moments of χ, restricted to the region contained in B are, respectively,

$$M = \int_{r<R} dr^3 \chi(\vec{r}), \tag{6.10}$$

$$\vec{M}_1 = \int_{r<R} dr^3 \chi(\vec{r})\vec{r}. \tag{6.11}$$

We can now estimate the normal direction, $\hat{\mathbf{n}}$, to S, as $\hat{\mathbf{n}} = -\vec{M}_1/|M_1|$ and the “thickness” h of B immersed in the bone, by solving $M = \pi h^2(R - \frac{h}{3})$. We can now derive, assuming that $\frac{h}{R} << 1$, and using Hertz’s contact theory [LL86], an expression for the total force, $\vec{F}_e$, exerted on the burr by the elastic deformation of the bone:

$$\vec{F}_e = C_1 R^2 (\frac{h}{R})^{\frac{3}{2}} \hat{\mathbf{n}}, \tag{6.12}$$

where C_1 is a dimensional constant, that describes the elastic properties of the material. Moreover, we can give an expression for the pressure, $\vec{P}(\vec{\xi})$, exerted by the burr on the point $\vec{\xi}$ of S:

$$\vec{P}(\vec{\xi}) = -\frac{3}{2\pi a^2}\sqrt{1 - \frac{|\vec{\xi}|^2}{a^2}}\vec{F}_e, \tag{6.13}$$

where $\vec{\xi}$ is measured from the center of S, see fig. 6.3(b), and a is the radius of the contact region. In Hertz’s contact theory, a can be estimated as

$$a = (C_1 R)^{\frac{1}{3}} F_e^{\frac{1}{3}}. \tag{6.14}$$

From equation 6.13, we can estimate the frictional force, $\vec{F_\mu}$, that the bone will oppose to the burr rotation:

$$\vec{F_\mu} = \mu \int_{\xi<a} d\sigma P(\vec{\xi}) \frac{\vec{r}(\vec{\xi}) \times \vec{\omega}}{|\vec{r}(\vec{\xi})||\vec{\omega}|}, \tag{6.15}$$

where μ is a friction coefficient, that links the frictional forces for unit area to the locally exerted pressure.

The total force that should be returned by the haptic feedback device is, therefore, $\vec{F_T} = \vec{F_e} + \vec{F_\mu}$.

Erosion modeling

We model the cutting of the burr by assuming that all the power spent by working against the frictional forces on a "contact surface" element $d\sigma$ goes toward the erosion of the bone material in contact with the surface. In other words, we equate

$$\mu P(\vec{\xi}) \omega r(\vec{\xi}) \left(1 - (\frac{\vec{r}(\vec{\xi}) \cdot \vec{\omega}}{|\vec{r}(\vec{\xi})||\vec{\omega}|})^2 \right) d\sigma = \alpha \phi(\vec{\xi}) d\sigma, \tag{6.16}$$

where α is a dimensional constant and $\phi(\vec{\xi})$ is the mass flux at the contact surface point $\vec{\xi}$. Using the mass flux ϕ one can update the position of the bone surface.

The formulas above have been written with the implicit assumption that the burr blades are very small with respect to the burr bit radius, and that their effect can be absorbed in the friction constant μ and in the "erosion constant" α. Even though this is, in general, false, and Hertz's theory is, strictly speaking, only valid for small elastic deformations, this formulation provides a computationally tractable, robust, expression for the response forces that, at least in the limit of small h, is physically reasonable.

6.4.2. Discretized description

Forces evaluation

In the simulator, the bone distribution is only known at the level of a volumetric grid discretized in cubic voxels. Eqs. (6.10, 6.11, 6.15) need, therefore, to be translated and re–interpreted.

A direct translation will transform integrals in sums over the voxels that have non null intersection with B. The evaluation of each voxel contribution is computationally complex, since it requires to find the intersections between B and the cube defining the voxel. To simplify matters, we are approximating the voxels with spheres of the same volume, centered at the voxel center, $\vec{c_i}$, whit the origin at the center of B. The radius of the voxel spheres, η, is, therefore, defined by $\frac{4}{3}\pi\eta^3 = \ell^3$, where ℓ is the length of the voxel side.

Using this approximation, it is trivial to derive simple formulas that express, in terms of the distance $d = |\vec{c_i}|$, the volume, ΔV, of the intersection region; the area, $\Delta\sigma$, of the "intersection surface" and the actual distance, r, from the center of the intersection surface to the center of B.

$$\Delta v(d) = \frac{\pi}{12}(d^3 - 6(R^2 + \eta^2)d + 8(R^3 + \eta^3) \tag{6.17}$$

$$- \; 3(\eta^2 - R^2)^2\frac{1}{d}) \tag{6.18}$$

$$\Delta\sigma(d) = \frac{\pi}{4}(2(\eta^2 + R^2) - d^2 - (\eta^2 - R^2)^2\frac{1}{d^2}) \tag{6.19}$$

$$r(d) = \frac{1}{2}d + \frac{R^2 - \eta^2}{2}\frac{1}{d} \tag{6.20}$$

The required integrals then become

$$M^* = \sum_i \Delta V(|\vec{c_i}|)\chi_i \tag{6.21}$$

and

$$\vec{M_1^*} = \sum_i \Delta V(|\vec{c_i}|)\chi_i \frac{r_i}{d_i}\vec{c_1}. \tag{6.22}$$

To estimate the friction force, $\vec{F_\mu}$ we convert the area integral (6.15) in

$$\vec{F_\mu} = \mu \sum_i \Delta\sigma(|\vec{c_i}|)P(\vec{\xi_i})\frac{\vec{c_i} \times \vec{\omega}}{|\vec{c_i}||\vec{\omega}|}, \text{with } \vec{\xi_i} = \frac{r_i}{d_i}(\vec{c_i} - \frac{(\vec{\omega} \cdot \vec{c_i})}{\omega^2}\vec{\omega}). \tag{6.23}$$

The power spent by the frictional forces on a voxel is then

$$\mu P(\xi_i)\omega r_i(\vec{\xi_i})\left(1 - (\frac{\vec{c_i} \cdot \vec{\omega}}{|\vec{c_i}||\vec{\omega}|})^2\right)\Delta\sigma_i = \alpha\phi_i\Delta\sigma_i, \tag{6.24}$$

where ϕ_i is the mass flux per unit surface coming out of voxel i, via surface $\Delta\sigma_i$. To evaluate P we use formula (6.13), where for a we use the "effective" radius of the contact surface $a^* = \sqrt{2Rh - h^2}$.

Erosion modeling

Using the fluxes ϕ_i we can now erode the voxels in the intersection region. In our current implementation, we associate a 8 bit counter with each voxel, representing the voxel density, and decrease it by a value proportional to the “assumed” amount of removed mass, $\Delta M_i = \Delta t \Delta \sigma \phi_i$, where Δt is the time step of the simulation, and the mass, M_i, contained in the voxel i. The bone material in the temporal bone area has a morphological structure that ranges from compact bone, e.g., close to the outer skull surface, to a porous, “trabecular”, consistency. The porous scale ranges from few millimeters down to scales well beyond the resolution of the medical imaging devices. In our model, the sub-scale modeling of the trabecular structures is absorbed in a voxel dependent erosion constant α. cutting sequence performed in the mastoid region.

6.4.3. Sample–Estimate-Hold Interface

A direct transmission of the computed forces to the haptic device is, in the case of “almost rigid” contacts, usually plagued by mechanical instabilities. The typical solution for this problem is the introduction of an artificial, “virtual”, coupling between the haptic device and the virtual environment [Col94, AH99].

In our system, we use a *sample–estimate–hold* approach [ESJ97] to remove the excess energy injected by the standard zero–order hold of force employed by the haptic device drivers. With this technique, we compute the force that is sent to the haptic device based on the previous zero–order representations produced at regular intervals by our burr–bone interaction model. This new value of force, when held over the corresponding sampling interval, approximates the force–time integral more closely than the usual zero–order hold [ESJ97].

6.5. Results

The performance of the prototype model is sufficient to meet timing constraints for force-feedback, even though the computational platform is made only of affordable and widely accessible components. We currently use a volume of 256x256x128 cubical voxels (0.3 mm side) to represent the region where the operation takes place. The force–feedback loop is running at 1 KHz using a 5x5x5 grid around the tip of the instruments for force computations. The computation needed for force evaluation and bone erosion takes typically 20μs, and less than 200μs in the worst case configuration.

In the following we will report on a series of experiments done using the prototype described above.

6.5.1. Force Evaluation

Figure 6.4 shows the reaction of the virtual bone against burr penetration. The computations are done in absence of erosion, $\alpha = \infty$, and using the actual force evaluation kernel of the force–feedback loop.

Figure 6.4(a) illustrates the “elastic” response of the material, measured in units of C_1R^2, as a function of the burr tip penetration depth measured in units of the burr bit radius R. Figure 6.4(b) illustrates the “frictional” response of the material, with $\mu = 1/2$ and for different angles $\theta, \theta = 30°, 60°, 90°$, between the surface normal and $\hat{\omega}$. The strength of F_μ increases for increasing $\sin(\theta)$. The knees in the F_μ curves correspond to the intersection of the burr bit with a deeper bone voxel layer.

Figure 6.5 shows the reaction of the virtual bone, again in runs with $\alpha = \infty$, to a sliding motion of the burr bit, immersed at a depth of $R/4$, over a flat bone surface. Fig. 6.5(a,b) show, respectively, the “elastic” and the “frictional” force response of the material, measured in units of C_1R^2, as a function of the distance traveled along the plane measured in R units. The pair of curves in each figure correspond to a sliding motion over a bone surface aligned along, respectively, one of the voxel discretization axis, and a plane with normal $[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$. The fluctuations in the force values are due to the “voxel sphere” approximation used to compute F. The difference in the wavelength of the fluctuations is a factor of $\sqrt{2}$ as expected.

6.5.2. Bone erosion

Figure 6.6 illustrates a “free–hand” experiment where bone is eroded by a polishing movement. The movement is similar to the one described in the previous subsection, with a sliding speed of about 10mm/sec, and $\alpha = 3.1 \times 10^6 \text{mm}^2/\text{sec}^2$. Figure 6.6(a) shows the depth of the burr below the surface level as a function of time, while fig. 6.6(b) reports the components of the force contributions and the total force applied to the haptic display during the movement. The lower line is the friction force $\vec{F_\mu}$, the middle line is the elastic force $\vec{F_{el}}$, and the upper line is the total force $\vec{F_{tot}}$.

6.6. Conclusion and Discussion

This chapter presented a haptic implementation of a bone cutting burr, that it is being developed as the main component of our training system for temporal bone surgery. We use a physically motivated model to describe the burr–bone interaction, that includes haptic forces evaluation, the bone erosion process and the resulting debris. The current implementation, directly operating on a voxel discretization of patient-specific 3D CT and MR imaging data, is efficient enough to provide real–time feedback on a low–end multi–processing PC platform. Next chapter will describe the adaptive implementation of the haptic model, designed and implemented in order to meet timing constraints when using burr with bit diameter more than 4mm.

6.7. Bibliographical notes

Most of the contents of this chapter were taken from papers [AGG$^+$02c, AGG$^+$03c], in which we discuss the haptic and visual simulation of a bone cutting burr, that can be considered the principal component of our training system.

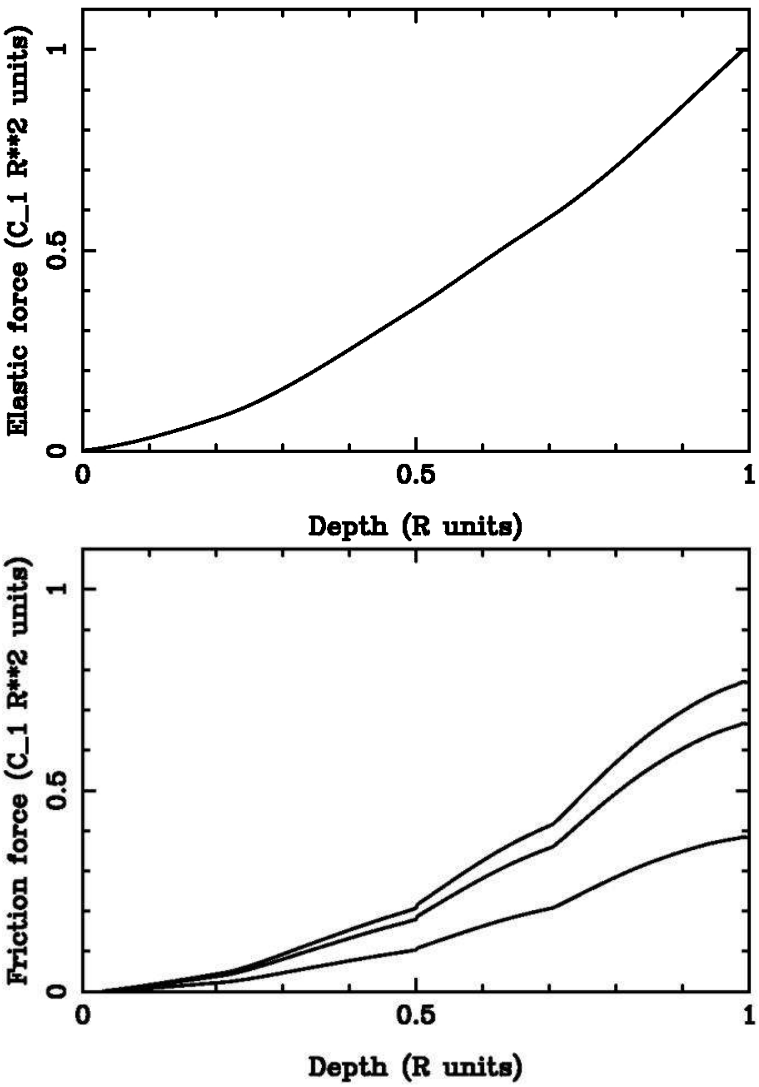


Figure 6.4.: **Virtual bone reaction against burr penetration.** Elastic and frictional responses, with $\mu = 1/2$ and for different angles $\theta, \theta = 30°, 60°, 90°$, between the surface normal and $\hat{\omega}$.

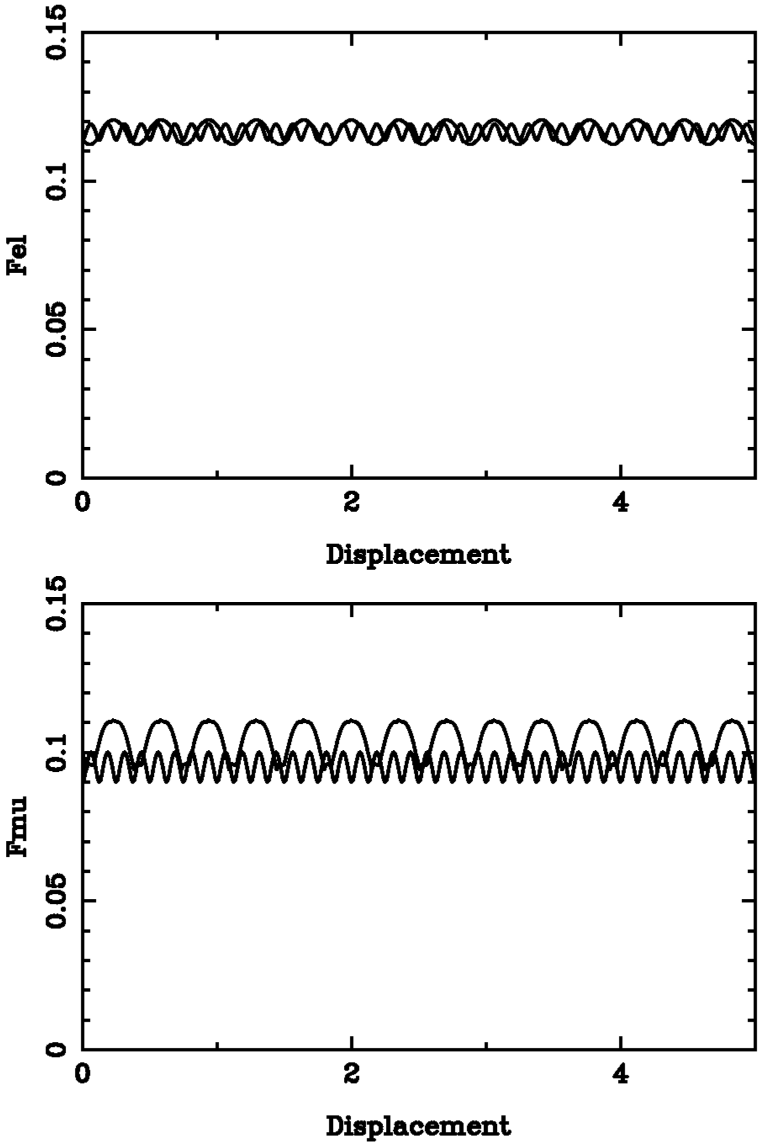


Figure 6.5.: **Sliding motion, constrained experiment**. Respectively, the elastic and frictional reaction of the flat surface of virtual bone to the sliding motion of a burr bit immersed at a depth of $R/4$.

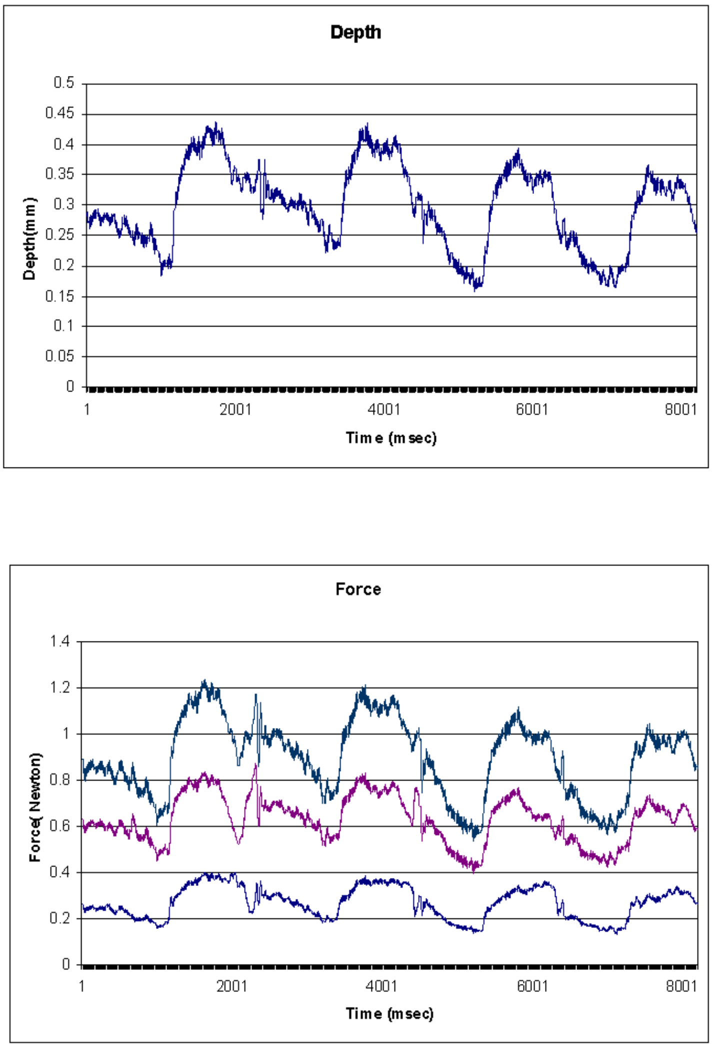


Figure 6.6.: **Bone erosion, polishing movement.** A “free–hand” experiment where bone is eroded by a polishing movement. Fig. (a) shows the depth of the burr below the surface level as a function of time. Fig. (b) reports the components of the force contributions and the total force applied to the haptic display during the movement.

7. Adaptive Techniques for Real-Time Haptic Simulation

In this chapter, we discuss adaptive techniques for providing real-time haptic feedback during the virtual bone dissection simulation. The adaptive technique exploits a multi-resolution representation of the first two moments of the bone characteristic function to rapidly compute contact forces and determine bone erosion. Thanks to our approach, we are able to simulate in real-time a wide range of operating conditions, such as initial cortex burring with large burr tips (up to 5 mm of radius), and deeper burring with accumulation of debris.

7.1. Introduction

Accurate and fast burr–bone interaction simulation is a key enabling technology in the development of a training simulator. It has to include burr–bone contact detection, bone erosion and generation of haptic response. The human perceptual requirements of a simulator impose very stringent constraints on performance, making bone dissection simulation a technological challenging task. In this chapter, we discuss adaptive techniques that trade simulation quality with speed in order to meet real-time constraints. The haptic component exploits a multi-resolution representation of the first two moments of the bone density to rapidly compute contact forces and determine bone erosion. The rest of the chapter is organized as follows. An overview of the related work is presented in section 7.2. Our techniques for real-time simulation of burr-bone interaction are briefly introduced in section 7.3. Section 7.4 discusses the preliminary results obtained.

7.2. Related work

Burr-bone interaction simulation builds on techniques developed for interaction with volumetric models. Early systems for interactively adding and removing material from a voxelized scalar representation were developed by Galyan and Hughes [GH91] and Wang and Kaufman [WK95]. These systems used a voxelized representation of both the editing tool and the data. Material removal was modeled as a simple Boolean operation, and there was no attempt at replicating the physical effect of tool/material contact. Avila and Sobjeraiski [AS96] extended the approach by adding repulsion forces, and used a force–feedback system to control a carving tool. Since contact detection and removal requires traversal of all voxels touched by the tool, the performance constraints of haptic feedback limited the system to model only small tools. Our previous work refined these techniques, specializing them to burr/bone interaction [AGG$^+$02c]. We introduced a physically based contact and erosion model loosely based on Hertz contact theory. The actual bone erosion is implemented by decreasing the density of the voxels that are in contact with the burr in a manner that is consistent with the predicted local mass flows. The method complexity scales, however, with the cube of the burr tip radius, imposing important limitations on the surgical tool size.

A number of authors have presented techniques based on probing a voxelized environment with surface point samples [GASF94, KCY93, MPT99b, HK97]. These techniques easily support free-form tool shapes, and, since they do not sample the interior of the tool tip, they are faster than their volumetric counterparts. They are however prone to severe aliasing artifacts, and are essentially limited to a one-voxel deep penetration[FCG00].

A few multi-resolution approaches have been proposed to speed-up volume sculpting operations, while avoiding the problems of surface point sampling approaches. Baerentzen [Bae98] proposed an octree-based volume sculpting system, that stores at each node local volume density, for quickly skipping empty regions, as in classic volume rendering applications. More recently, Frisken et al. [FPRJ00] proposed a resolution adaptive approach based on multi-resolution distance fields. The basic idea is to store at each node the Euclidean distance from the closest surface instead of the local density. This representation considerably speeds up collision detection and response, but requires a costly distance propagation step after model modification. Otaduy and Lin [OL03] introduce a sensation preserving simplification algorithm for faster collision queries between two polyhedral objects in haptic rendering. Given a polyhedral model, we construct a multi-resolution hierarchy using filtered edge collapse, subject to constraints imposed by collision detection. The resulting hierarchy is then used to compute fast contact response for haptic display. The computation model is inspired by human tactual perception of contact information. As in [Bae98], we also use an octree-based representation of the data, which is continuously kept up-to-date during simulation. Instead of storing local density at each node, we store the first two moments of the bone characteristic function, which provide us information on the local mass distribution, without imposing the model update overhead of a distance based approach. This information is exploited for multi-resolution force computation.

7.3. Burr-bone interaction and haptic feedback

A detailed mechanical description of the cutting of material by a rotating burr is complicated because it involves: the tracking of the continuously changing free surface of the material being cut; the impact of the burr blades on the surface; the resulting stress distribution in the material; and the consequent plastic deformation and break–up. In the general engineering context these problems are solved by using experimentally determined curves, but, for the specific case of bone burring, there are no publicly available data. Furthermore, in the specific context of haptic feedback, one cannot apply the standard methods found in the mechanical engineering literature for the simulation of milling. In fact, an haptic feedback system is driven by an open–loop controller that needs to rapidly evaluate a reasonable response force for arbitrary tool penetrations.

To circumvent these complications, we have developed a simplified model, originally described in [AGG$^+$02c], based on a limited number of parameters that are, at the moment, tuned by trial and error following the opinion of expert surgeons as feedback. The basic assumption underlying our model is that the burr bit is moving relatively slowly with respect to the time scale of the haptic feedback loop and that one can estimate the elastic forces exerted by the bone by geometrically characterizing the region of bone intersected by an idealized sphere representing the burr tip.

Specifically, we model the burr bit, B, with a sphere of radius R centered at $\mathbf{R}_b$, and consider the first two moments of the bone mass density, $\rho(\mathbf{r})$, contained in B.

$$m_0 = \int_{r<R} dr^3 \rho(\mathbf{r}), \mathbf{m}_1 = \int_{r<R} dr^3 \rho(\mathbf{r})\mathbf{r}. \tag{7.1}$$

The direction of the local normal, $\hat{\mathbf{n}}$, to the bone surface can then be estimated as $\hat{\mathbf{n}} = -\mathbf{m}_1/|m_1|$, and from the amount of mass contained in B, m_0, we can derive an effective "penetration depth" h as the smallest positive solution of

$$m_0 = \pi\rho_0 R^3 (\frac{h}{R})^2 (1 - \frac{h}{3R}) \tag{7.2}$$

where ρ_0 is the “solid” bone reference density.

We can now write an expression for an effective force $\mathbf{F}_e$, that is supposed to model the elastic response of the bone to the impinging burr.

$$\mathbf{F}_e = c_e R^2 (h/R)^{3/2} \hat{\mathbf{n}}, \tag{7.3}$$

where c_e is a dimensional constant, that, as far as this model is concerned, describes the elastic properties of the material. In the limit of $h/R << 1$, eq. (7.3) is consistent with Hertz’s contact theory [LL86].

Typical burr radii are between 1 mm and 5 mm, while the typical speed at which the burr bit is moved is $<$ 100 mm/s [AGG$^+$02a]. Given that the haptic device acquisition period is 1 ms, the burr bit will typically move a distance of the order of a few percents of its radius. Therefore, it is reasonable to compute interaction forces by checking collisions after the fact, rather than trying to predict them in advance.

For a given response force F_e^* one can invert eq. (7.3) to obtain, assuming that $h/R << 1$,

$$m_0^* \approx \pi \rho_0 R^{1/3} (F_e^*/c_e)^{4/3}. \tag{7.4}$$

Hence, the amount of computational work needed to resolve, say, a zero force threshold increases only slowly with the burr radius. On the other hand, in typical burr usage one applies a force on the burr so that it will have an instantaneous erosion surface that scales as R^2. Since the contact surface, S, of the burr with the bone, again for small h, is proportional to hR, this corresponds to a mode of operation where the force applied by the user on the burr is adjusted to maintain h roughly proportional to R, $h \approx \alpha R$. With this assumption,

$$\frac{dF_e}{d\,m_0} = \frac{3}{4}\frac{c_e}{\pi\rho_0}\frac{1}{R}(\frac{h}{R})^{-1/2}\frac{1}{1-\frac{h}{2R}}, \tag{7.5}$$

and

$$F_e = c_e R^2 \alpha^{3/2}, \tag{7.6}$$

therefore

$$\frac{1}{F_e}\frac{dF_e}{d\,m_0} \approx \frac{3}{4}\frac{1}{\pi\rho_0}\frac{1}{R^3\alpha^2}. \tag{7.7}$$

For a given accepted error ratio β in the haptic force, $\beta = \Delta F_e/F_e$, we can estimate the accepted error for m_0, Δm_0, to be

$$\Delta m_0 \approx \beta \alpha^2 R^3. \tag{7.8}$$

Therefore, in this mode of operation, we can maintain the relative error in force estimation constant at a small computational cost even for increasing burr radius R. In fact, we are allowed to increase linearly with R the discretization scale, ℓ, used in computing the integrals in eq 7.1.

In the following, we will describe a computational method that exploits these observations to compute $\mathbf{F}_e$ with a computational cost that grows slowly with R and is well within the time constraints, 1 msec total for force estimate and bone erosion, imposed by the haptic feedback device.

The new method completely overcomes the limitations to small burr sizes of the technique used in [AGG$^+$02c].

7.3.1. Multi-scale spatial description

The integrals requested by eq. 7.1 can be easily computed using a multi–resolution volumetric description of the region of interest.

We partition the volume of interest using an octree, with the leaves of the octree that directly refer to the scene voxels, and the coarsest level to the whole scene. In an initialization phase, starting from the leaves, we precompute, for each octree block, I, the local values of m_0^I and $\mathbf{m}_1^I$. The zeroth moment of the mass contained in block I is simply the sum of its values at the block children $\{I,k\}$, $m_0^{\{I,k\}}$. To compute $\vec{m}_1^I$ we use the center of mass decomposition rule

$$\mathbf{m}_1^I = \sum_k [\mathbf{r}_I^k m_0^{\{I,k\}} + \mathbf{m}_1^{\{I,k\}}], \tag{7.9}$$

where $\mathbf{r}_I^k$ is the vector that goes from the center of block I to the center of its child k.

The algorithm used to estimate m_0 and $\mathbf{m}_1$ is then the following. At each haptic cycle, we descend the octree until we find blocks that are either fully contained or partially intersecting the burr sphere. If they are fully contained, we add their contribution to m_0 and $\mathbf{m}_1$; if they are partially intersection, we compare the block size with ℓ and if it is larger we refine; otherwise, we add the partial volume contributions

$$\begin{aligned} \Delta m_0 &= \frac{\Delta V}{V_I} m_0^I & (7.10) \\ \Delta \mathbf{m}_1 &= \frac{\Delta V}{V^I}(\mathbf{R}_b - \mathbf{R}_c)(m_0^I + \mathbf{m}_1^I \cdot (\mathbf{R}_c - \mathbf{R}^I) \\ &\qquad + O((\ell/R)^2), & (7.11) \end{aligned}$$

where ΔV is the volume of the region of intersection between block I and the sphere, $\mathbf{R}_c$ is the position of the center of mass of the latter intersection, and $\mathbf{R}^I$ is the position of the center of block I. In the current implementation of the algorithm, both ΔV and $\mathbf{R}_c$ are approximated by replacing the block with a sphere of equal volume (see [AGG+02c] for details).

Therefore, at the cost of a minor computational overhead in the precomputing and update (see below the discussion on erosion), we are able to estimate m_0 and $\mathbf{m}_1$ with a computational cost that grows as most as h^2R/ℓ^3. Moreover, the availability of the precomputed $\mathbf{m}_1^I$ moments allow us to estimate the contribution of partially overlapping blocks at a higher order to what would have been possible using only m_0^I. We are thus allowed to use larger values for ℓ.

7.3.2. Multi-scale erosion

Erosion, i.e. material removal in response to burring, is modeled as a position dependent erosion rate described by f, an erosion shape function,

$$\frac{d\rho(\mathbf{r})}{dt} = \alpha f(r/R)\rho(\mathbf{r}); \tag{7.12}$$

where, again, $\mathbf{r}$ is measured from the center of B, and α is an appropriate dimensional constant. f is constrained to have a maximum at $r/R = 0$ and to be null for $r/R > 1$. In the previous chapter, erosion was modeled by assuming that all the power spent by working against the frictional forces on a "contact surface" element of the bone would have gone toward the erosion of the bone material on the surface (Reye's hypothesis). The resulting expression for the local mass derivative was, however, rather complex and computationally expensive. Eq. (7.12) provides essentially comparable results at a much lower computational cost.

From the point of view of the implementation, in our model the bone is described as a collection of voxels, each one containing up to 255 values of bone occupation. To accommodate for a wide range of erosion rates using only 8 bits, we convert the rate of erosion given in Eq. (7.12) to a probability that the value of the voxel at position $\mathbf{r}$ will be reduced by one at next time step. A Russian roulette scheme is then used for deciding whether to fully erode a bit (i.e. remove 1/255th of the mass of a full voxel) or not.

To find the voxels that should be eroded, we integrate the following modifications to the octree descent algorithm introduced above. When we identify a block as contained in the burr, we descend down to all the leaves and erode the voxels using the probabilistic version of Eq. (7.12). When the block is instead only partially contained in the burr, we continue recursion until we find completely contained sub blocks and then proceed as above. If we reach a leaf which is only partially contained, the erosion probability is scaled by the overlap fraction before testing for erosion. In descending the octree we keep track of the number of voxels touched while visiting a node children. If it changes, we perform an update of the node value from its children values using the same scheme used for octree construction (i.e. pulling moment updates from octree leaves up to the root).

7.3.3. Other contributions to the haptic response

Together with the elastic force $\mathbf{F}_e$ defined in eq. (7.3), we also compute a frictional force, $\mathbf{F}_\mu$, that is supposed to model the friction forces that oppose burr rotation when the latter is in contact with the bone material; and an impact force, $\mathbf{F}_i$ that can be thought as what would be the response of the bone material if it were modeled as a collection of unconnected point masses swept by the moving burr sphere.

$$\begin{aligned} \mathbf{F}_\mu &= c_\mu R^2 (h/R)(\mathbf{m}_1/m_0) \times \omega & (7.13) \\ \mathbf{F}_i &= -(c_i R^2)(h/R)\mathbf{V} & (7.14) \end{aligned}$$

where we have introduced ω, representing the burr angular velocity vector, and $\mathbf{V}$ the velocity of the burr center.

7.4. Results

In the following we will report on a series of experiments done using the prototype described above.

7.4.1. Multi-resolution Force Evaluation

Figure 7.1(a) shows the reaction of the virtual bone against burr penetration, using different burr-sizes and different accuracy parameters. The computations are done in absence of erosion, $\alpha = 0$ in equation 7.12, and using the actual force evaluation kernel of the force–feedback loop with a volume composed of cubical voxels with 0.3 mm side. The figure shows the "elastic" response of the material when using two different burr sizes ($R = 1.0mm, R = 5.0mm$), which correspond to a standard polishing burr tip and a large initial burring tip. The force has been computed using the mono-resolution algorithm, as well as three different accuracy settings of the multi-resolution algorithm, corresponding to ($\ell = 0.1R, \ell = 0.3R, \ell = 0.5R$). The graphs clearly show that the mono-resolution and the multi-resolution version of the algorithm are in agreement. In figure 7.1(b), we report how the the relative error with respect to the reference mono-resolution solution changes with penetration. As it can be seen from the figure, it is typically of the order of few percents or below. The oscillations in the curves are due to resonances between the burr position and the octree grid used to compute the forces.

In figure 7.2, we report the wall clock time required by the force computation kernel to compute the forces of figure 7.1(a). Each subfigure shows the wall clock time required for different values of the burr radius and for different resolution scales. For $R = 1$mm, as expected, there is no appreciable between the mono-resolution results and the multi-resolution ones for $\ell = 0.1R, \ell = 0.3R$, while the $\ell = 0.5R$ is faster. For $R = 3$mm $\ell = 0.1R$ is still of the same order of the voxel size, 0.25mm, while the $\ell = 0.3R$ and $\ell = 0.5R$ are now clearly faster than the mono-resolution case.

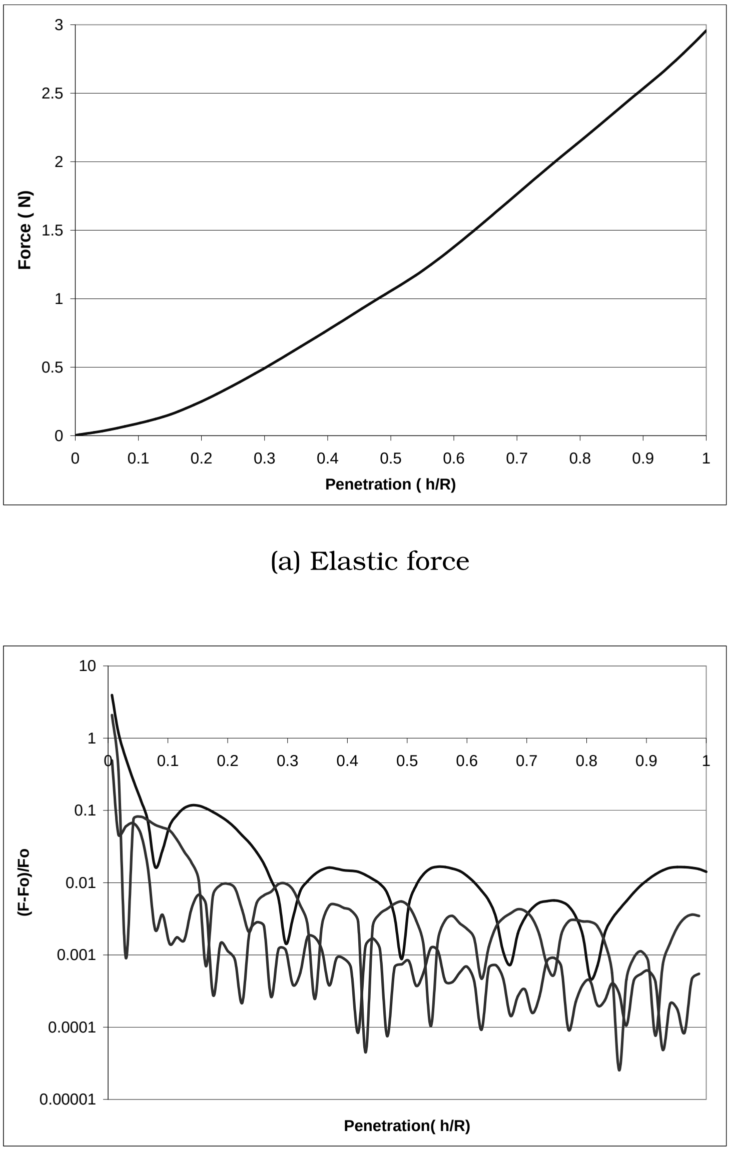
In figure 7.3, we show the growth of the computational cost for a given resolution scale and different radius values, $R = 1, 3, 5$mm. The figures show the growth of the computational cost for a given resolution scale and different radius values, $R = 1, 3, 5$mm. It is clear from the figures that the mono-resolution algorithm is limited to $R < 2$mm, the computational cost of the multi-resolution algorithm grows very slowly with R. and always easily meets the 1 ms haptic feedback time constraint.

7.5. Conclusion and Discussion

In this chapter we have presented adaptive techniques for providing real-time haptic and visual feedback during simulation of a bone cutting burr. The adaptive technique exploits a multi-resolution representation of the first two moments of the bone characteristic function to rapidly compute contact forces and determine bone erosion. The haptic simulation is being developed as a component of a training system for temporal bone surgery. Thanks to our adaptive techniques, we are able to simulate in real-time a wide range of operating conditions, such as initial cortex burring with large burr tips (up to 5 mm of radius), and deeper burring with accumulation of debris.

7.6. Bibliographical notes

This chapter reviews and expands the contents of paper [AGG$^+$03b], where we presented an adaptive bone dissection haptic model, exploiting a multi-resolution representation of the first two moments of the bone characteristic function to rapidly compute contact forces and determine bone erosion.



(a) Elastic force

(b) Relative error

Figure 7.1.: Virtual bone reaction against burr penetration and relative error in force evaluation introduced by the multiscale algorithm.

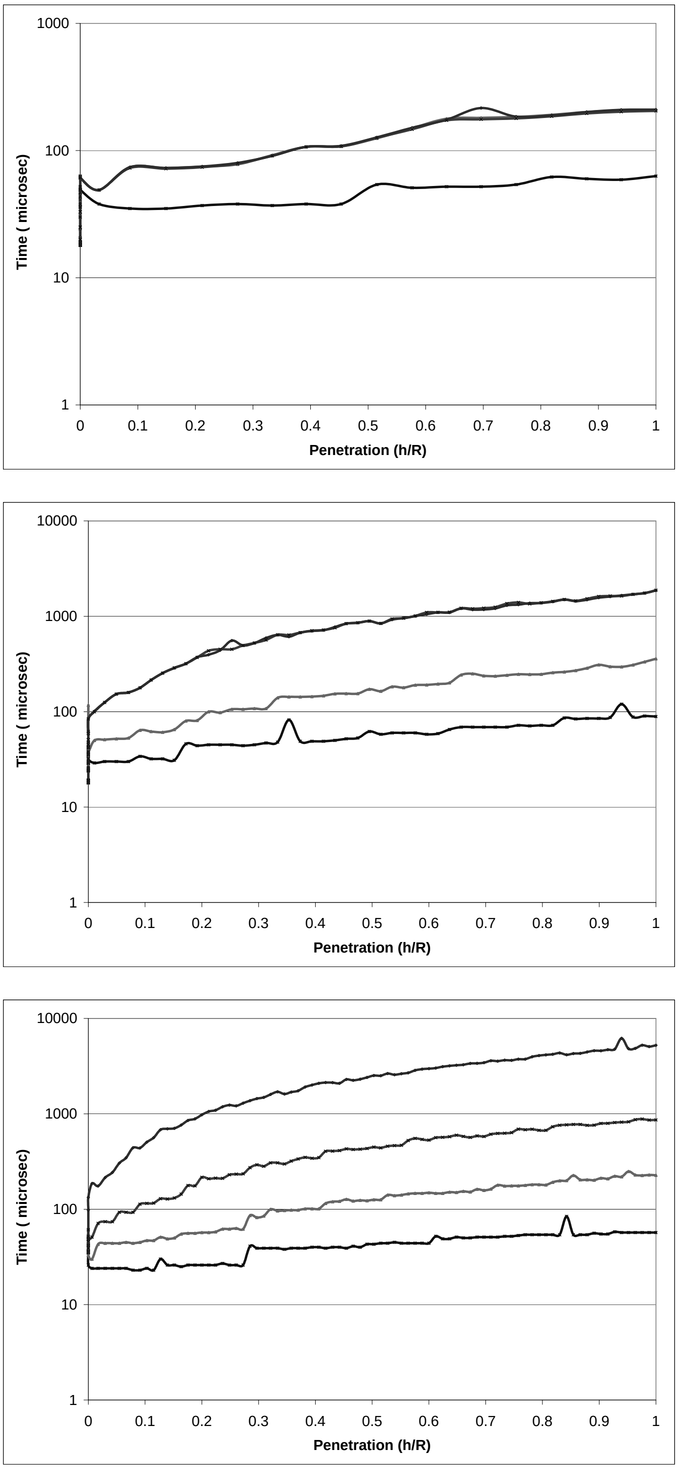


Figure 7.2.: Time required to compute the forces of figure 7.1(a) compared by radius. From top to bottom, considered radii are, respectively $R = 1$mm, $R = 3$mm and $R = 5$mm.

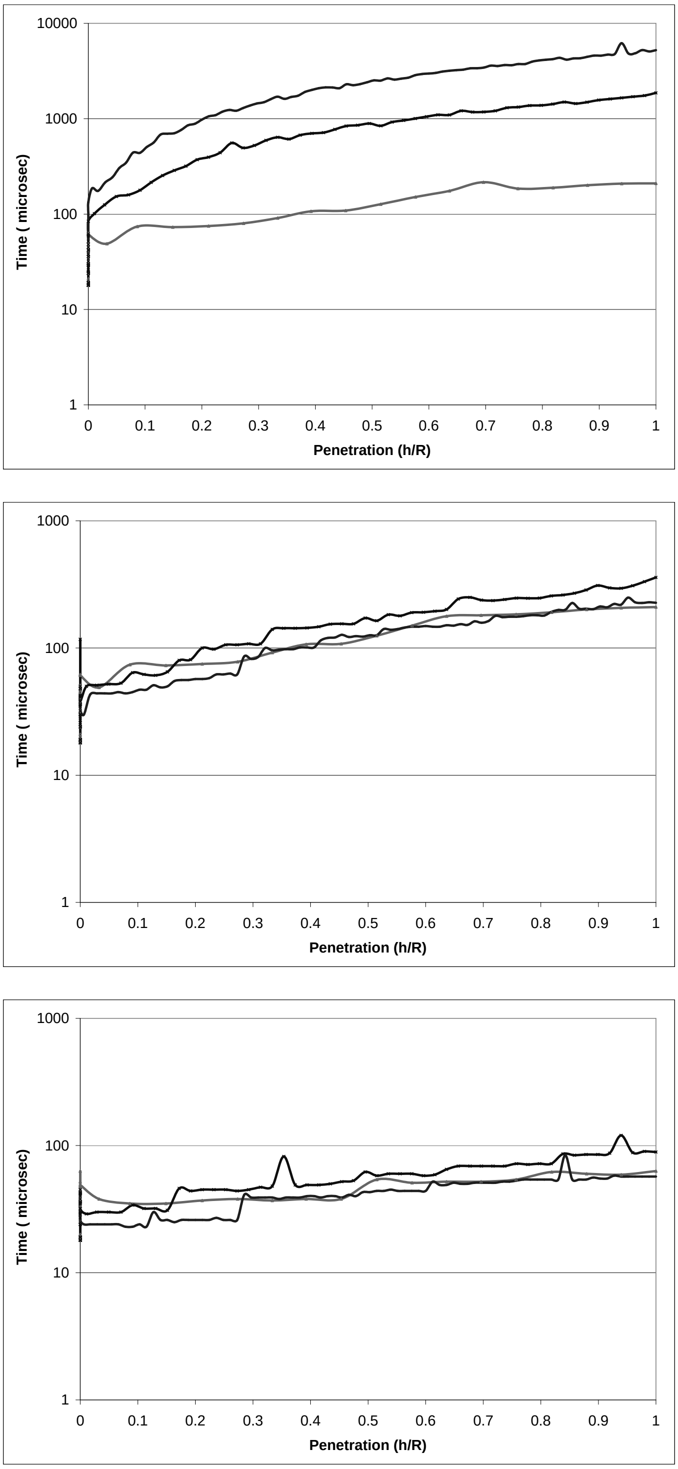


Figure 7.3.: Time required to compute the forces of figure 7.1(a) compared by resolution scale. From top to bottom, the resolution scales are $\ell = 0$, $\ell = 0.3R$, $\ell = 0.5R$.

8. Experimental Validation and Calibration

In surgical training simulators, in order to realistically reproduce the operating conditions, haptic models are commonly tuned with a trial and error approach, by following expert users indications. While subjective input from selected end users is encouraging, it would be of extreme interest to compare virtual results with direct forces measurements obtained by drilling actual samples. Since, to our knowledge, there are no available data on the subject in literature, we decided to define an experimental setup and measurement procedures. This chapter describes the strategy we followed to collect experimental data that will be used to develop and validate our physically based contact and bone erosion model. The experimental set–up consists of a force–controlled robot arm holding a high–speed rotating tool and a contact force measuring apparatus. Contact forces and tool positions are measured during the milling process under various operating conditions. Preliminary experimental results are shown.

8.1. Introduction

In bone dissection virtual simulation, force restitution represents the key to realistically mimicking a patient–specific operating environment. The force is rendered using haptic devices controlled by parameterized mathematical models that represent the bone–burr contact. The choice of parameters is usually done by trial and error following the opinion of expert surgeons as feedback. As an alternative, one can try to determine the relevant parameters by comparing the predictions of the model used with the results of experiments in controlled environments. In this chapter we describe the strategy we followed to collect experimental data that will be used to develop and validate the physically based contact and bone erosion model presented in chapter 6. The experimental set–up consists of a force–controlled robot arm holding a high–speed rotating tool and a contact force measuring apparatus. Contact forces and tool positions are measured during the milling process under various operating conditions. Preliminary experimental results are shown. The chapter is organized as follows: section 8.2 reviews the state of the art, relatively to the experimental validation of haptic models, sections 8.3 and 8.4 describe the home-built experimental facility that we employed to characterize the burring operations, while section 8.6 and section 8.7 show the preliminary experimental results that we obtained.

8.2. Related work

The most relevant work related to the experimental validation of haptic systems was presented by Okamura and others [OIN$^+$03]. They developed the Haptic Scissors, a device that creates the sensation of cutting in virtual environments and has two degrees of freedom of motion and force feedback, one for cutting (single blade rotation) and one for translation. They used filtered data from cutting biological tissues to create haptic recordings of the cutting experience. The use of haptic recordings rather than reality-based models, is simple and computationally efficient, but can be only applied to systems with a small number of degrees of freedom[CGO02]. Brower and others [BUB$^+$01] developed devices to measure tissue properties under extension and indentation, as well as to record instrument-tissue interaction forces. They tested their system on porcine abdominal tissues, and concentrated their attention to differences between in-vivo and ex-vivo measurements. Our work is characterized by a physically based contact model of stiff materials, the use of patient specific data, and the focus on validating the haptic model with experimental data. To our knowledge, such data is not currently available in the literature.

8.3. Experimental setup

In order to validate the model described in chapters 6 and 7, we developed and built an experimental equipment. This experimental system may measure and record contact forces between burr and material during controlled movements, and it contains the following components:

- a three degrees of freedom composite robot arm , capable of controlled movements with a precision under $100\mu m$;
- a mini drill MINICRAFT, model MB150, commonly used for surgical training;

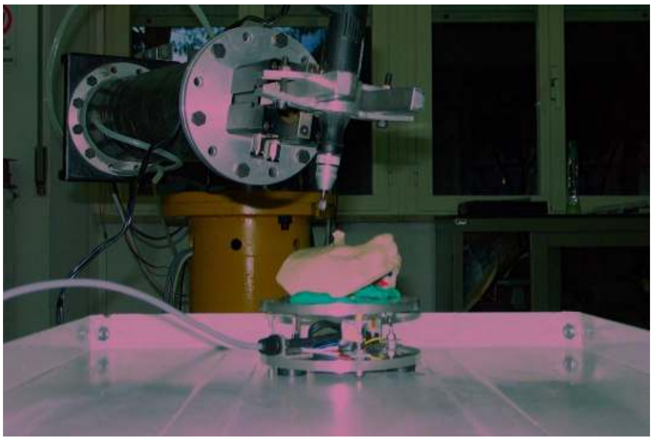


Figure 8.1.: **Experimental setup**: a robot arm equipped with a high-velocity burr, moves along the vertical direction, while a load cell records contact forces to feed a velocity loop controller.

- a dynamo-metric platform upon which the bone specimen is placed.

Figure 8.1 shows the experimental system in action, with the burring tool moving against the sample, and the dynamo-metric platform.

8.4. Dynamo-metric platform

8.4.1. Architecture

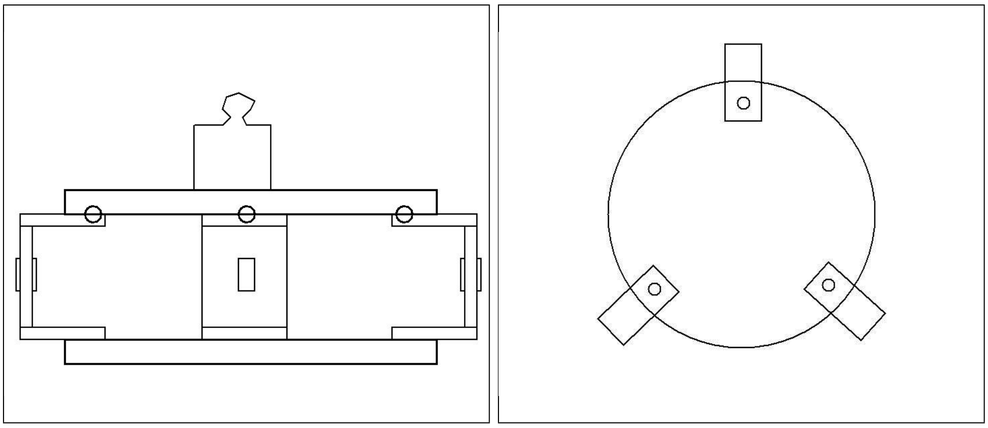


Figure 8.2.: **Load cell architecture**: top and lateral views. The cell is made of two aluminum plates and three strain sensors.

We realized two platforms owing the same architectures, that is represented in figure 8.2. Each of them contains the following pieces:

- a aluminum base plate, and a measure one, with the same diameter and width, respectively 125 mm and 7 mm;
- three C shaped strain sensors, that were obtained from an aluminum profile.

The three sensors are fixed to the base and placed as to form the vertexes of an equilateral triangle. This setup has been chosen in order to permit an adequate load subdivision between the sensors. Hence, sensors are connected to the platform through spherical couplings.

8.4.2. Strain sensors

	Cell A	Cell B
length(l)	15 mm	15 mm
height(h)	25 mm	50 mm
width(w)	2.0 mm	1.2 mm
arm(b)	28 mm	40 mm

Table 8.1.: **Sensor geometry parameters**

The strain sensors used for the realization of the load cells are built by applying electrical strain gauges to C shaped alumina's pieces, with geometrical parameters reported in table 8.4.2. The electrical strain-gauges are realized from Kiowa [AGG+03a], and they provide temperature compensation as well as a gauge factor 1.98. They are applied vertically on opposite sides of the piece. If we take into account the sensor positions and directions, a vertical load applied to the platform is equally subdivided between the three sensors, because of the spherical coupling. Now, in order to compute the load to each sensor, we need to derive the linear relationship between the load itself and the strain measured by strain-gauges.

Given a sensor like that represented in figure 8.3, a load P, applied to the spherical joint, causes the following reactions, that are felt by strain-gauges:

1. compression normal action $N = -P$,
2. bending action $M = Pb$, where b is the sensor arm.

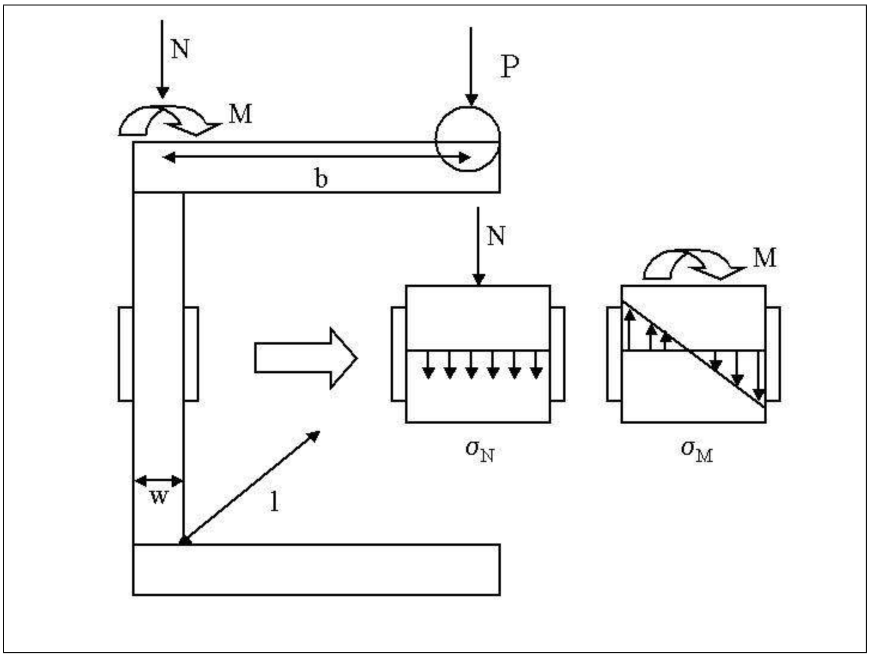


Figure 8.3.: **Sensor strains**: a load P causes compression and bending stresses that are felt as strains by strain-gauges.

If we just consider a infinitely small width in the strain-gauges neighborhood, the strain diagrams are like those reported in figure 8.3, so that felt stresses are:

$$\sigma_{in} = \sigma_N + \sigma_M \tag{8.1}$$

$$\sigma_{out} = \sigma_N - \sigma_M \tag{8.2}$$

where

$$\sigma_N = \frac{N}{A} = -\frac{P}{lw} \tag{8.3}$$

$$\sigma_M = \frac{M\frac{w}{2}}{J} = 6\frac{Pb}{lw^2} \tag{8.4}$$

according to Saint-Venant theory [TG70].

From previous relationships, we are able to derive the strains, by applying the Hooke elasticity law:

$$\epsilon_N = -\frac{P}{lwE} \tag{8.5}$$

$$\epsilon_M = 6\frac{Pb}{lw^2E}. \tag{8.6}$$

Since a Wheatstone half-bridge circuit can measure the strain differences between internal and external strain-gauges [BV75], the overall sensor strain measured is

$$\epsilon = \epsilon_{in} - \epsilon_{out} = 2\epsilon_M = 12\frac{Pb}{lw^2E}$$

, that is proportional to applied load. Considering the aluminum Young modulus and Poisson ratio, and the geometric parameters in table 8.4.2, the sensors built for load cell A and B, have the following sensitivity factors:

$$K_\epsilon^A = \frac{\partial \epsilon}{\partial P} = 80\frac{\mu\epsilon}{N} \tag{8.7}$$

$$K_\epsilon^B = \frac{\partial \epsilon}{\partial P} = 320\frac{\mu\epsilon}{N}. \tag{8.8}$$

These factors were also empirically verified. Each sensor was calibrated by using a set of known weights, in order to have the maximum accuracy.

8.5. DAQ System Configuration

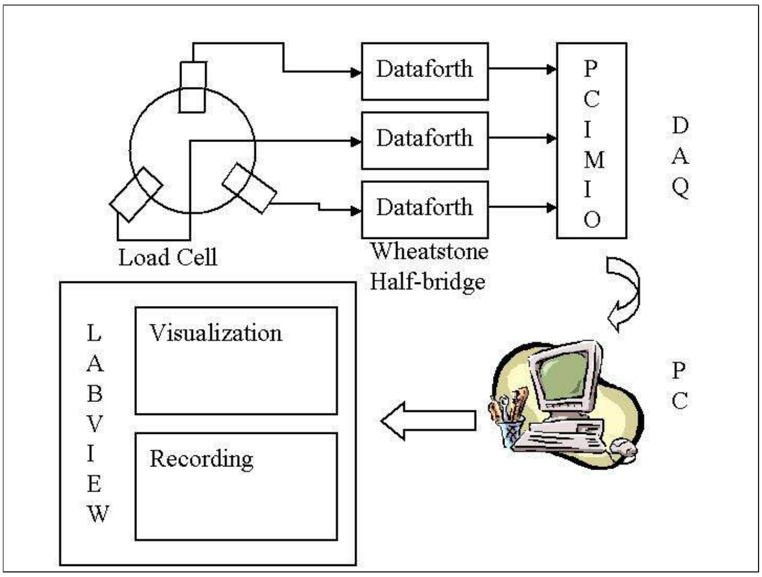


Figure 8.4.: Load cells signals DAQ system

In order to measure, visualize and record strain signals, strain gauges are connected to DATAFORTH signal-conditioning modules. Voltage signals are then acquired through a National Instruments PCI-MIO-16E4 DAQ system, as represented in figure 8.4. A LabView Virtual Instrument, designed in G programming language [JJJ01], visualizes voltage signals from DATAFORTH modules, and convert them in one force signal, by applying sensitivity factors empirically derived. Since DAQ card resolution is 12 bit, corresponding to a $5mV$ threshold, the overall sensitivity is improved by employing digital filtering. Specifically, an RMS 7-th order digital filter is applied to input voltage signals, while a median one is applied to the output force signal. Digital filtering obviously reduces the system bandwidth, but since we are interested to low-frequency dynamics, a 100 Hz bandwidth is to be considered adequate for our scopes.

8.6. Experimental sessions

The model described in chapters 6 and 7 by eq. (6.12) and eq. (7.12) is, undoubtedly, over-simplistic being mainly motivated by practical computational reasons. Its behavior is controlled by two constants, c_e and α whose value should be determined by the material modeled. While in a first moment we tuned these constants to be consistent with the subjective surgeons experience, here we try to define a value for them based on direct experimental measures. Specifically, we have selected a simple reference experiment, the vertical descent – at constant applied force – of a burr into the material, that we perform both in a real experimental set-up and its virtual analogue. Under the assumption that our simplified model captures some of the main features of the real system, we then fix the parameters of the virtual model by a non–linear fit of the simulated to the experimental data.

We perform this procedure to fit to human petrous bone data, Pettigrew Plastic Temporal Bones [Pet] a synthetic resin model of the temporal bone widely used in surgical training, and, for reference purposes, PVC K70 resin.

That experimental procedure we employed is based upon a constant–force feedback control system applied to the arm robot [VK93, Sch88]. Specifically, experimental sessions are conducted by applying a constant force to the tool, and by measuring and recording the tool displacement-time characteristics. From the full collection of the displacement-time characteristics having force values as parameter, a force-feed law can be obtained. Moreover, the same experimental sessions can be used to calibrate the haptic system, by deriving the parameters of the model, that depend on the material and tool characteristics. Actually, experimental displacement-time characteristics can be compared with the theoretical ones obtained by computing tool displacements when a constant force is applied to the theoretical model. Hence, the model calibration can be achieved by tuning parameters until, for each force value, experimental displacements

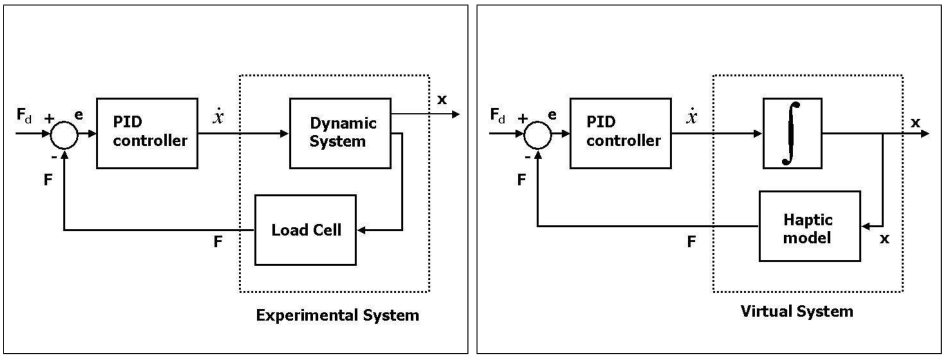


Figure 8.5.: **Real Control System vs Virtual Control System**: the same PID controller and the same kind of force-feedback control is applied to the real system, and to the theoretical model, in order to compare the results.

match theoretical ones.

Figure 8.5 shows the architecture of the real experimental control system, and the virtual one, where the robot arm–burr–bone system is substituted with the theoretical model.

In figure 8.6 we show a plot of the typical force, position and velocity measurements done during a run in PVC. After the impact of the burr on the material, the burr proceeds at an essentially constant velocity until it is well inside (more that one third of its radius) the material volume. We will use this velocity, see next subsection, to characterize the behavior of the material at that level of applied force.

8.7. Results

In figure 8.7 we report our preliminary measurements of the initial penetration velocity of the burr on a human temporal bone sample for different levels of applied constant forces. The scatter of the measured data is due to the inhomogeneous nature of the material.

The solid line plotted in the same figure represents the data generated by the virtual analogue experiment with parameters fitted to minimize the square sum of the velocity descent difference between the experimental and the virtual data. Since the selection of the parameters results from the non-linear fitting of procedural generated data, we were not able to derive direct estimates of the confidence interval of the fitted parameters. As an indicative measure of the latter, we show the two lines with dots corresponding to erosion values $\alpha = 0.077$ and $\alpha = 0.097$.

In figure 8.8 we report our preliminary measurements of the initial penetration velocity of the burr on samples of Pettigrew Plastic Temporal Bones [Pet], widely used in surgical training. Such synthetic models are currently a valid alternative to cadaveric exercises. Using such models, the complete temporal bone, for example, can be fully dissected using standard theater equipment.

Again, as in the case of the real temporal bone there is scatter in the data, and the dotted lines correspond to erosion values $\alpha = 0.106$ and $\alpha = 0.126$. In the case our preliminary results indicate that best-fit parameter values are $c_e = 2.145$ and $\alpha = 0.087$ for temporal bone, and $c_e = 1.504$ and $\alpha = 0.116$ for Pettigrew plastic temporal bone replies.

As a reference, we acquired homologous data for PVC K70 resin. These resulted in parameter values $c_e = 0.462$ and $\alpha = 0.296$, as shown in figure 8.9.

Notably, the expert selected values we have previously used are consistent with the values measured for the actual temporal bone, but not with the Pettigrew Bone result. See, however, the results of the psychophysical experiments described in chapter 13.

8.8. Conclusion and Discussion

This chapter described the strategy we followed to collect experimental data that will be used to develop and validate our physically based contact and bone erosion model. The experimental set–up consists of a force–controlled robot arm holding a high–speed rotating tool and a contact force measuring apparatus. Contact forces and tool positions are measured during the milling process under various operating conditions. Preliminary experimental results showed that expert selected values we have previously used are consistent with the values measured for the actual temporal bone, but not with the Pettigrew Bone result.

8.9. Bibliographical notes

The contents of this chapter are taken from reference [ABG$^+$04], that provides a preliminary report on our work on the tuning of a temporal bone surgical simulator using parameter values derived from experimental measurements, and on the comparison between these results and the previously used domain expert assigned values.

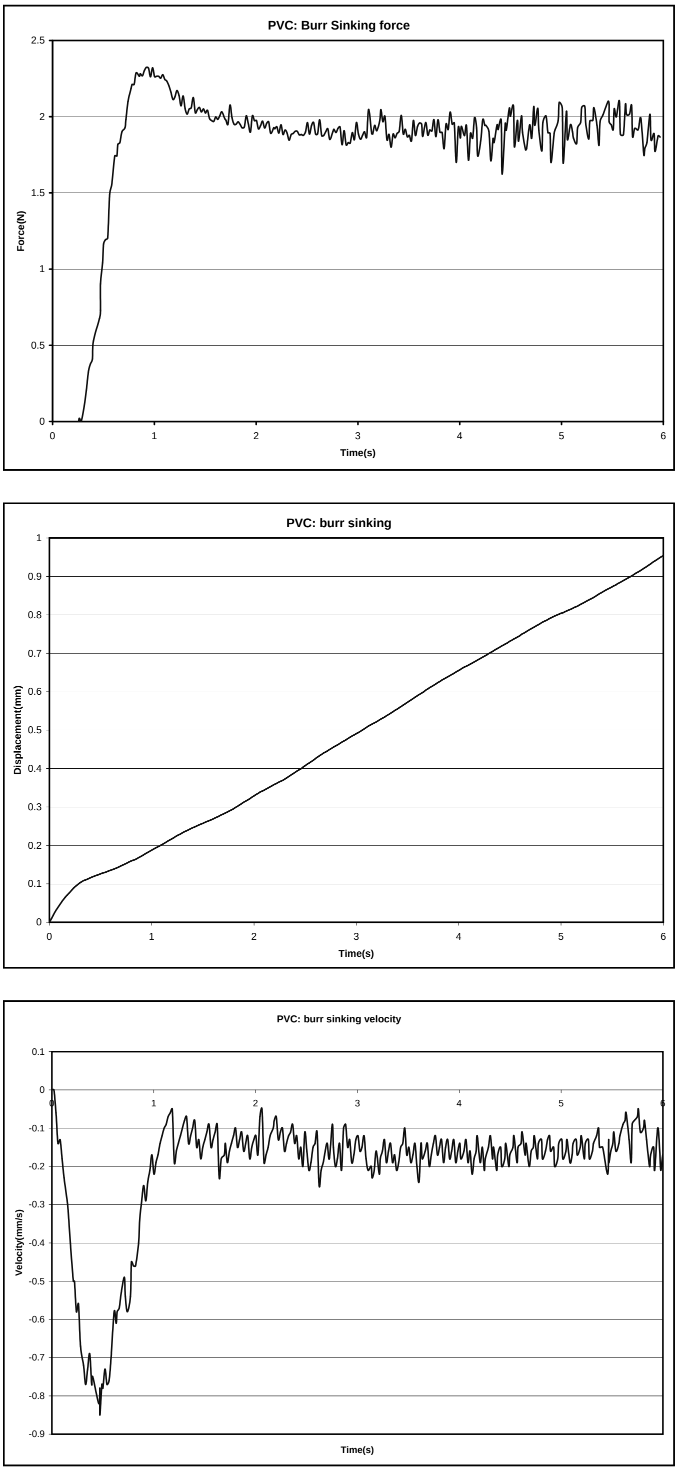


Figure 8.6.: Typical force, position and velocity plots recorded by our experimental system while burring on a block of PVC. Note the impact of the burr on the material and the velocity stabilization at time 1s.

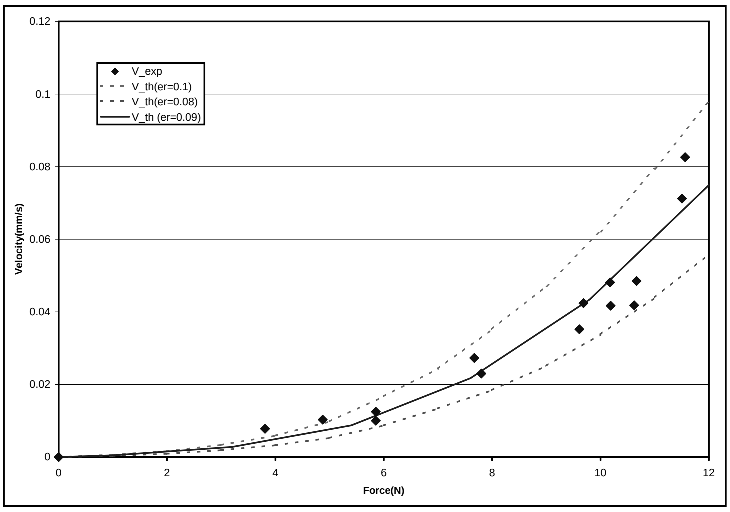


Figure 8.7.: **Haptic model fitting with bone experimental data**: lines with dots represent the confidence interval on the erosion factor parameter($\alpha = 0.077$ and $\alpha = 0.097$).

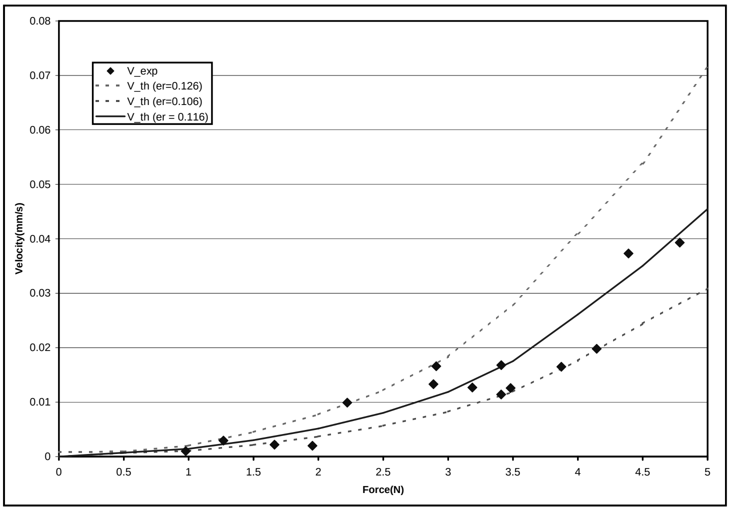


Figure 8.8.: **Haptic model fitting with Pettigrew plastic temporal bone experimental data**: lines with dots represent the confidence interval on the erosion factor parameter ($\alpha = 0.106$ and $\alpha = 0.126$).

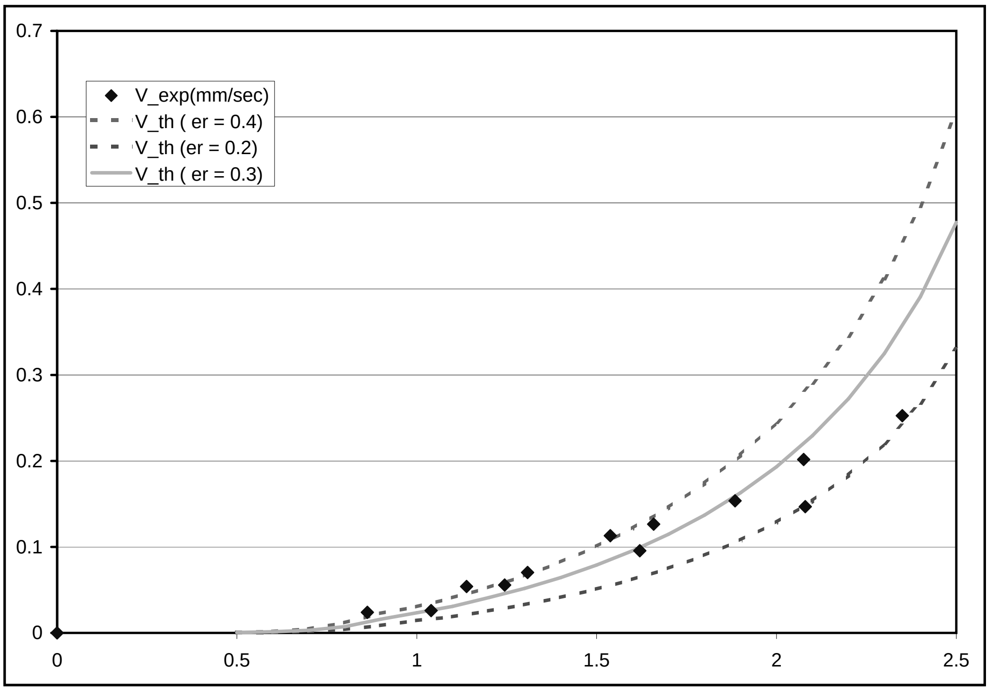


Figure 8.9.: **Haptic model fitting with PVC K70 resin experimental data**: lines with dots represent the confidence interval on the erosion factor parameter ($\alpha = 0.196$ and $\alpha = 0.396$).

Part III.

Visual System

As for the visual effect of the drill on the surface of the bone, the user requirements task analysis (see chapter 4), highlighted that the graphical process must simulate drill site obscuration by bone dust paste, because its absence would reduce the importance placed by a trainee on the need for regular irrigation and suction. Realistic and meaningful bleeding is a perennial problem for VR researchers. We have concluded that, visually, the actual drill representation needs only be quite simple, and it is felt that representing the spinning of the cutter or diamond burr is unnecessary. What is considered necessary, from a functional standpoint, is an effective collision detection mechanism which not only copes with increased resolution as the virtual drill proceeds deeper into the temporal bone, but is also capable of generating error states when (for example) a large burr is inserted into a narrow drill site. As for the nature of the technology required for displaying drill, drill site, bone, and so on, there is no conclusive evidence or support for the premise that the use of a stereoscopic system will aid performance in this case. Binocular viewing systems are deployed in the operating theatre and used by surgeons, and so binocular imaging should be available to the simulator. However, the wearing of any form of stereoscopic display, such as a headmounted display or liquid crystal shutter glasses should be avoided. The surgeon or trainee does not want to use cumbersome eyewear that is not necessary for carrying on the real procedure. We make the hypothesis that, if the simulation achieves a reasonable level of fidelity, then the combination of high-resolution images and haptic feedback will, more than likely, suffice. This part of the dissertation focuses on the components of the visual system. Specifically, chapter 9 describes our real-time direct volumetric rendering system based upon Phong lighting model, while chapter 10 focuses on the dust/fluid dynamics component developed to simulate secondary "obscuring" effects.

9. Hardware-Accelerated Volume Rendering

In most of direct volume rendering applications, one of the greatest problem is related to how to render dynamically changing volumes. This feature is often required by virtual reality systems, but current available hardware and traditional rendering approaches are not able to reach real-time performances. We developed a direct volume rendering approach, based upon texture mapping hardware and the new NVIdia OpenGL combiner extensions, that enable fast global volume reloading. Our technique allows to obtain high quality direct volume rendering on standard PC graphics hardware, at interactive frame rates. The global reloading feature simplifies the use of this volume renderer inside physically-based simulators.This chapter describes our real-time direct volumetric rendering approach based upon Phong lighting model. The technique relies on the absorption plus emission optical model, and it performs real-time sampling, mapping and optical rendering integration, by just employing OpenGL 1.2 primitives. Then, Phong lighting model computation, based upon opacity gradient, is performed through Register Combiner extensions.

9.1. Introduction

Interactive volume rendering has become an invaluable technique to visualize 3D scalar data for a variety of applications in engineering, science and medicine. New imaging technologies like computed tomography (CT), magnetic resonance imaging (MRI), and 3D ultrasound generate 3D data sets describing the interior of volumes. A volume is regarded as a cube which is partitioned by a regular grid into small cubes called voxels. The imaging techniques measure some physical feature of each voxel (e.g., the spin of atoms) and generate a three-dimensional array of numerical values describing property of the volume. Typical sizes of the array range from 128x128x64 to 256x256x256. Whereas two-dimensional images can readily displayed on computer or TV displays, it is not obvious how volumetric data should be presented. One can select any plane through the volume and display it. However, this does not give a three-dimensional impression. To view the whole volume within one view, voxels are made more or less transparent according to the related numerical value where values of interest are emphasized and other are made transparent. This allows interior voxel of interest to become visible. Classical approaches to volume rendering use ray tracing which is very time-consuming. Therefore it cannot be performed in real time – necessary for interactive applications – except by using special dedicated hardware architectures. Texture-based volume rendering achieves similar results in real time by use of standard texture and blending functionality of modern graphics computers. In surgical training systems, volume updates are made for each frame, especially in bone drilling tasks, where surgical burrs continuously erode bone. In this context, a volume renderer requires continuous data updates. Hence, interactive frame rates, given the limitations of graphics hardware available, are currently not obtainable. In this chapter we propose a solution for this problem. The volume renderer component that we designed and implemented for our temporal bone surgery training virtual system exploits a texture-based approach. An inherent deficiency of the original texture-based rendering algorithms was that lighting could not be included whereas ray-tracing algorithms can deal with lights. We describe techniques for realistic shading and lighting using computer graphics hardware. In particular, we discuss multi-texturing and high quality local illumination methods based upon NVidia OpenGL per-fragment combiner extensions. These results are then combined with a combined texture and stencil buffer masking technique in order to reach an higher framerate. The rest of the chapter is organized as follows. Section 9.2 details the related work. In Section 9.3, the capabilities of current PC rasterization hardware are described. Since our approach exploits multi-texturing and multi-stage rasterization, these features are explained in detail. In Section 9.4 the basic ideas of a texture based volume rendering approach are explained. Section 9.5 adapts the optical model for fast rendering of shaded slices to PC rasterization hardware. Section 9.6 shows the texture zoom approach that we implemented to enhance frame rate performance without altering rendering quality. In Section 6.5 the results of our study are evaluated.

9.2. Related work

There is a variety of different visualization approaches for scalar volumes in multiple application scenarios. Recent approaches are categorized into indirect methods, such as isosurface extraction [UK88, LC87], and direct methods, that immediately display the voxel data. We will focus on interactive direct methods. The basic idea of using object-aligned slices to substitute trilinear by bilinear interpolation was presented by Lacroute and Levoy, although the original implementation did not use texturing hardware. For the PC platform, Brady et al. have presented a technique for interactive volume navigation based on 2D-texture mapping. More recently, Mueller et al. used image based techniques to improve the performance of volume ray-casting. Volume rendering with 2D textures is more complex and does not provide as good results as 3D textures, but can be used on any OpenGL implementation. The problem with 2D textures is that the data slice polygons can't always be perpendicular to the view direction. Three sets of 2D texture maps are to be created, each set perpendicular to one of the major axes of the data volume. These texture sets are created from adjacent 2D slices of the original 3D volume data along a major axis. The data slice polygons must be aligned with whichever set of 2D texture maps is most parallel to it. In the worst case, the data slices are canted 45 degrees from the view direction. The more edge-on the slices are to the eye, the worse the data sampling is. In the extreme case of an edge-on slice, the textured values on the slices aren't blended at all. At each edge pixel, only one sample is visible, from the line of texel values crossing the polygon slice. All the other values are obscured. However the resulting images often contain visual artifacts caused by the lack of spatial interpolation. This chapter discusses new rendering techniques that significantly improve both performance and image quality of the 2D-texture based approach. Ertl and others [EKE01, RSEB$^+$00, WE98] showed how multi-texturing capabilities of modern consumer PC graphics boards can be exploited to enable interaction with volumes. The image quality of the 2D-texture based implementation can be greatly enhanced by performing real trilinear interpolation. This is achieved without loss in performance by interpolating intermediate slices using multi-textures. Volume rendering techniques that exploit the 2D-texturing hardware of PC graphics boards usually produce images that contain visual artifacts. The basic 2D-texture based approach is to decompose the volume into a set of object-aligned slices. The necessary trilinear interpolation can then be reduced to a bilinear interpolation which can be efficiently computed by standard texturing hardware. However, when zooming closely on a small detail inside the volume data, which is often done in medical applications, the missing trilinear interpolation is strongly visible. Due to the large number of trilinear interpolations that must be processed in order to produce image results of high quality, the availability of direct volume ren-

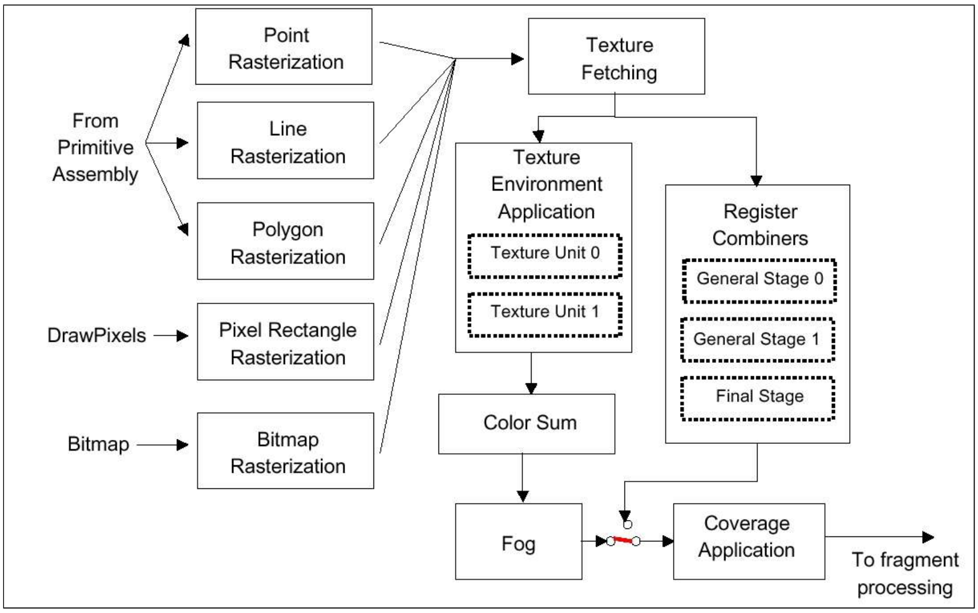


Figure 9.1.: **The extended NVidia OpenGL pipeline.** The register combiners replace the standard OpenGL per-fragment environment.

dering has yet been restricted to high-end workstations and special purpose graphics hardware. Although there is a clear trend toward standard PC hardware as visualization platform, the application of interactive hardware-accelerated approaches is still limited. If 3D-textures (OpenGL 1.2) are supported by hardware, it is possible to render slices parallel to the image plane with respect to the current viewing direction. This means that if the viewing matrix changes, these view-port-aligned slices must be recomputed. To obtain correct visual results with this approach opacity values must be scaled according to the distance between two adjacent slices in direction of the viewing ray. Like in the 3D-texture based approach, scaling the values linearly with a constant factor as an approximation has lead to good visual results.

9.3. The OpenGL Register combiner extension

The GeForce and Quadro register combiners functionality provides a configurable (but not programmable) means to determine per-pixel fragment coloring [Kil00]. When enabled, the register combiners replace the standard OpenGL per-fragment texture environment, color sum, and fog operations with a greatly extended mechanism for coloring fragments. Figure 9.1 shows this data flow. With multi-textured

OpenGL, filtered texels from each texture unit are combined with the fragment's current color in sequential order. The color sum and fog stages immediately follow the texture environment in a fixed sequence. The register combiners expose a sequence of general combiner stages that terminate in a final combiner stage that outputs an RGBA color for the fragment. The register combiners support the following functionality:

- Multiple combiner inputs are available in each combiner. Primary and secondary (specular) colors, the fog color and factor, two constant colors, and two spare inputs are available.
- Computations in each general combiner stage use a signed numeric range from [-1,1] instead of an unsigned [0,1] range.
- The numeric range of each input is mapped and possibly clamped using one of eight distinct input mappings. These input mappings provide conversions from unsigned to signed numeric ranges, negation, half-biasing, and unsigned inversion.
- The RGB and alpha portions are configured and processed independently.
- Each general combiner stage outputs three distinct outputs for both the RGB and alpha portions.
- Possible outputs are products of inputs, a sum of products of inputs, 3-element vector dot products of RGB inputs, or a mux of products of inputs.
- Each stage writes its outputs to a set of registers that be-come the inputs for the subsequent stage. Unwritten register values carry forward from one stage to the next.
- A special final combiner stage combines the final register values into a final RGB and alpha result for each fragment.

9.4. Direct volume rendering optical model

Direct volume rendering is typically regarded as a technique for generating images directly from volume data, without intermediate reconstructions. Images are computed through a **transfer function**, that maps required optical properties (color, emission and absorption coefficient) onto a given physical measure. Normally volumes are seen as suspension particles or clouds or blobby gels. The typical rendering effects are obtained by integrating material optical properties (absorption/emission/reflection functions) along light ray paths. The optical properties are mathematical models defining material response to light. Hence, rendering algorithms can be computationally expensive. Their complexity strongly relies on integration approximation steps, on light ray density, and on optical model accuracy. Hence, if the principal target of the rendering system is to allow interactivity, image quality needs to be sacrificed. Hence, in order to reduce the computation and rendering time, integral coarse approximations and simplified optical models are usually employed. On the other hand, wheninteracting with virtual environments, humans are very sensitive to *temporal fidelity*. Temporal fidelity can be evaluated by measuring the frame-rate and the latency time. The minimum acceptable frame-rate is strictly related to user tasks, and varies from 10 fps to 50 fps. The latency is the time passed between an user action and the correspondent user reaction. Some experiments were lead on distributed cooperative virtual systems [PK99] and on virtual systems with view-point tracking [AHJ$^+$01], and showed that virtual handling difficulties are proportional to system latency since from 40 ms. If latency is greater than 300 ms, tasks become almost impossible to complete. In [EYAE99, ERSS99], it is highlighted that users are sensible to latency variations under 30 ms. During last 15 years, various optical model for direct volume rendering were proposed [Max95]. The model generally adopted in interactive systems is the one named **absorption plus emission**. The intensity variation of a light ray passing through a section S and an infinitesimal length ds, is given by the following differential equation:

$$\frac{ds}{Is} = g(s) - \tau(s)I(s) = C(s)\tau(s) - \tau(s)I(s), \tag{9.1}$$

where $\tau(s)$, $I(s)$, $C(s)$ are, respectively, the extinction coefficient, the light intensity and the specific reflect/emitted intensity in s. The integral of $C(s)$ between 0 and s defines the resultant light intensity in s. The solution of equation 9.1 can be approximated with the *particle model* [Max95], by assuming that voxel width is the integration step, and that $\tau(n)$ and $C(n)$ are constant inside the voxel itself:

$$\widetilde{C}_N = \sum_{n=1}^{N} C[n]\alpha[n] \prod_{m=1}^{n-1} (1 - \alpha[m]), \tag{9.2}$$

where $\widetilde{C}_N$ is the associated color [Bli94] resulting from N volume slices, while

$$T_N = \prod_{n=1}^{N} (1 - \alpha[n]) \tag{9.3}$$

represents the equivalent transparency, of n layers passed through by the light ray. If only associated colors are employed, the 9.2 becomes:

$$\widetilde{C}_N = \sum_{n=1}^{N} \widetilde{C}[n] \prod_{m=1}^{n-1} (1 - \alpha[m]) \tag{9.4}$$

and, if $t[m]$ is the slice transparency given by $(1 - \alpha[m])$ difference, then [Sab88]:

$$\widetilde{C}_N = \sum_{n=1}^{N} \widetilde{C}[n] \prod_{m=1}^{n-1} t[m]. \tag{9.5}$$

Considering the step n in the integration process, the slice n contribution to total color $\widetilde{C}_N$ is given by:

$$\widetilde{C}_N = \widetilde{C}[n] + (1 - \alpha[n])\widetilde{C}_{N-1}, \tag{9.6}$$

where $\widetilde{C}_{N-1}$ is the accumulated associated color from previous iterations, when accumulation is back-to-front, or:

$$\widetilde{C}_N = \widetilde{C}[n] + (1 - \alpha[n])\widetilde{C}_{N-1}, \tag{9.7}$$

where α_{N-1} is the overall opacity of the first N-1 slices, when accumulation is front-to-back. Normally, $g(s)$ can be substituted with the *scattering function* $g(s, \bar{\omega})$ [Max95], which takes into account also the light direction. In general,

$$g(s, \bar{\omega}) = E(s) + L(s, \bar{\omega}) \tag{9.8}$$

where $E(s)$ is the adirectional emissive or reflective radiation component and $L(s, \bar{\omega})$ is the directional component, function of the view direction $\bar{\omega}$. Making the following assumptions:

- a generic voxel is reached by radiation $i(s, \bar{\omega}')$ without any obscuring or absorption from other voxels
- the radiation $i(s, \bar{\omega}')$ comes from a directional source
- $p(s, \bar{\omega}, \bar{\omega}')$ is the *phase function*, representing the reflection factor in view direction $\bar{\omega}$

the term $L(s, \bar{\omega})$ in equation 9.8 may be written as

$$L(s, \bar{\omega}) = A(s)\tau(s)p(s, \bar{\omega}, \bar{\omega}')i(s, \bar{\omega}') \tag{9.9}$$

where $\tau(s)$ is the extinction coefficient and $A(s)$ is the reflection factor, also known as*particle albedo*. If we only consider Lambertian reflection, the phase function depends only on lighting direction $\bar{\omega}'$ and surface normal $\bar{n}$, that can be computed as gradient of the volume function f:

$$L(s,\bar{\omega}) = i(s,\bar{\omega}')A(s)\tau(s)max\left(\frac{\nabla f(s)}{\|\nabla f(s)\|}\cdot\bar{\omega}',0\right). \qquad (9.10)$$

Substituting equation 9.10 in equation 9.8, and assuming that term $E(s)$ is reflective, the associated color $\widetilde{C}[n]$ comparing in equation 9.6 may be expressed as:

$$\begin{aligned}\widetilde{C}[n] &= \widetilde{C}_a[n]l_a + \widetilde{C}_d[n]max\left(\frac{\nabla f[n]}{\|\nabla f[n]\|}\cdot\bar{l}_d,0\right)\\ &= \alpha[n]C_a[n]l_a \qquad (9.11)\\ &+ \alpha[n]C_d[n]\left\|\bar{l}_d\right\|max\left(\frac{\nabla f[n]}{\|\nabla f[n]\|}\cdot\frac{\bar{l}_d}{\left\|\bar{l}_d\right\|},0\right) \qquad (9.12)\end{aligned}$$

where $C_a[n]$, $C_d[n]$ and $\alpha[n]$ are the non-directional ambient reflective factor, the diffuse directional reflective factor and the voxel opacity, while l_a and $\bar{l}_d$ are the ambient light intensity and the light intensity coming from the directional source. In order to highlight surface details, we can employ an artifact, like weighting the opacity $\alpha[n]$ with a surface *strength*, evaluated as function of volume and his gradient: $S = h\left(f(s),\nabla f(s)\right)$ [DCH98, Lev88]. If we use the gradient modulus as *strength*, we have:

$$\widetilde{C}[n] = l_a\left\|\nabla f[n]\right\|\alpha[n]C_a[n] + \left\|\bar{l}_d\right\|\alpha[n]C_d[n]max\left(\nabla f[n]\cdot\frac{\bar{l}_d}{\left\|\bar{l}_d\right\|},0\right). \qquad (9.13)$$

Such a strength function, enables the visualization of boundary surface between tissues, and disables the visualization of parts with null gradient (like the internal parts of an object).

9.5. Interactive volume rendering approach

Although volumetric data is defined over a continuous 3D domain (R^3), measurements and simulations provide volume data as 3D arrays, where each of the scalar values that comprise the volume data set is obtained by sampling the continuous domain at a discrete location. These values are referred to as voxels (volume elements) and usually quantized to 8, 16, or 32 bit accuracy and saved as fixed point or floating point numbers. Hence volume data can be easily used as scalar texture images, without pre-processing. According to graphics hardware features, data is loaded as a 3D texture or as view-aligned 2D texture stacks. The integration process is executed front-to-back or back-to-front, by sampling the volume through texturing hardware. This process is like *throwing* as many rays as the number of pixels that cover the view window, and computing equations (9.6) e (9.7). For each step and for all pixels, graphics hardware accesses to texture and extracts the scalar value. This sampled value is converted, through a transfer function, to a color triple and an opacity value that are saved inside combiner input registers. Combiners may be programmed to implement equation 9.13, and return the associated color $\widetilde{C}_N$, that is then used during the blending process. Texture hardware can perform *multi-texturing*, so combiners are able to compute the gradient $\nabla f[n]$ needed to derive the associated color. In following subsections we provide more detail about this process.

9.5.1. Volume pre-processing

The only needed data pre-processing required in direct volume rendering is related to the format required by graphics hardware. The first constraint is about texture resolution. Although arbitrary texture sizes can be defined with the OpenGL EXT_texture_rectangle extension (but not in every graphics hardware), the general size is a 2-power number for each dimension. But, in general, in order to meet timing requirements, a volume resize is required according to the view window dimensions. Actually, the optimal ratio between volume resolution and window size, is such that a single voxel is projected to each screen pixel. That's why, when volume data is projected to a higher resolution window, the filling process is slowed down and no additional information is added to image, while when a too high resolution volume is rendered, a graphics memory bandwidth overload occurs without a sensitive quality increase. Another pre-processing phase aims to reduce the texel size. In many cases, such as 12 bit CT volumes, data can be coded with 8 bit without significant precision lost. In fact, since tissues distribution density is normally contained inside a sub-range of CT values, the coding scheme is generally a simple translation and, eventually, a scaling.

9.5.2. Sampling through texture mapping

Current consumer graphics hardware is based on an object-order rasterization approach, i.e. primitives (polygons, lines, points) are scan-converted and written pixel-per-pixel into the frame buffer. Since volume data do not consist of such primitives, a proxy geometry is defined for each individual slice through the volume data. Each slice is textured with the corresponding data from the volume. The volume is reconstructed during rasterization on the slice polygon by applying a convolution of the volume data with a filter kernel. The entire volume can be represented by a stack of such slices, if the number of slices satisfies the restrictions imposed by the Nyquist theorem. After texture loading and reconstruction, the rasterization process derives, for each projected pixel, the texture sampling position, and the texturing hardware extracts the correspondent density, by bi-linear or tri-linear interpolation of the closest texels; the resultant value is then mapped to an RGBA vector by the transfer function. In order to compute the surface gradient, four texture units are needed, that are used to sample the volume with offset dx, dy, dz, relatively to the central point, as described in algorithm 1.

9.5.3. Transfer function

The transfer function mapping, in our direct volume rendering approach, is obtained by exploiting the *glColorTable* primitive, that is commonly implemented in commodity graphics hardware. This function is used by compiling a look-up table, with the transfer function values, and by installing it inside the graphics memory. In the same time of sampling, texturing hardware performs the transfer conversion of density values to the RGBA colors contained inside the table. The color table contains associated colors instead of pure colors [WMG98, Bli94], in order to control the color interpolation error. The associated color employment has also beneficial effects to color accumulation process (cfr. equations 9.2 and 9.4). The color look-up table let users choose and calibrate the transfer function in real time; it can be computed and reloaded each time the user change some function parameters.

9.5.4. Opacity gradient computation

In lighting equation 9.13, the surface normal is related to gradient $\nabla f[n]$, and the modulus is regarded as surface *strength*. If $f[n]$ is used as opacity, instead of density, we can arbitrarily modify the surface appearance properties (opacity, width and consistence) by modifying the transfer function. Since OpenGL register combiners receive from texture hardware 4 opacity values $\alpha(p)$, $\alpha(p+dx)$, $\alpha(p+dy)$, $\alpha(p+dz)$, they are able to approximate the opacity gradient with forward differences:

$$\begin{aligned}\nabla_x &= \frac{\alpha(p) - \alpha(p + dx)}{dx}\\ \nabla_y &= \frac{\alpha(p) - \alpha(p + dy)}{dy}\\ \nabla_z &= \frac{\alpha(p) - \alpha(p + dz)}{dz}.\end{aligned}$$

Since combiners are SIMD arithmetic modules able to perform linear operations, it is relatively simple to derive forward differences and vector modules, but it is impossible to perform ratios and root extractions. Now the gradient norm computation involves a square root extraction, that needs to be approximated with a polynomial function. Since the number of available combiners is limited and many of them are used to compute the gradient components as well as the lighting equation, we can anly approximate with a quadratic function. The 2nd order polynomial is derived from a Taylor series evaluated in the neighborhood of an arbitrary point x_0 of interval]0, 1].

9.5.5. Voxel color computation

According to equation 9.13, the shading components are supposed to be the combination of an ambient component and a directional component emitted by a source along the volume z axis (slices normal). In this way, the dot product between the light direction and the opacity gradient is the component $\nabla_z f[n]$, and equation 9.13 is simplified as follows:

$$\widetilde{C}[n] = l_a \left\|\nabla f[n]\right\| \alpha[n]C_a[n] + l_d\alpha[n]C_d[n]max\left(\nabla_z f[n], 0\right). \tag{9.14}$$

But visual result of equation 9.14 is not fully satisfying: in fact only surface voxels contribute to pixel color, because strength becomes zero in tissue internal parts. This fact would be irrelevant if surfaces were consistent enough to completely mask the color of internal voxels. Anyway, low strength surfaces and small width walls let see the hollows produced by equation 9.14 (see fig. 9.2a). Hence, the optical model employed for internal volumes is different from that used exclusively for parts having non null gradient. The overall model is then defined by:

$$\widetilde{C}[n] = \begin{cases} \begin{array}{c} l_a \left\|\nabla f[n]\right\| \alpha[n]C_a[n] \\ +l_d\alpha[n]C_d[n]max\left(\nabla_z f[n], 0\right) \end{array} & if \quad \left\|\nabla f[n]\right\| > 0 \\ \\ \left(l_a + l_d\right)\alpha[n]C_a[n] & if \quad \left\|\nabla f[n]\right\| = 0 \end{cases}. \tag{9.15}$$

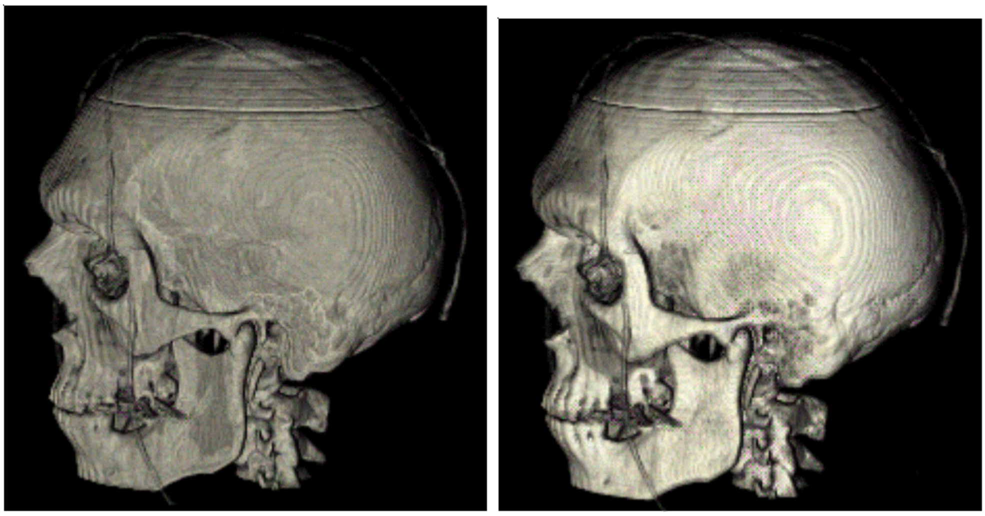


Figure 9.2.: Internal and external optical models.

With this artifact, the image quality is greatly improved during rendering of low density tissues or in these case of tissue subtle layers with high density (for example bone), as shown in figure 9.2b.

9.5.6. Opacity and color accumulation

The accumulation equations 9.6 e 9.7 can be efficiently implemented on all OpenGL graphics platform. The accumulation is obtained by simply enabling the alpha blending function (GL_BLEND) and by setting the *source* and *destination* blending factors, with (ONE, ONE_MINUS_SRC_ALPHA) for back-to-front composition and (ONE_MINUS_DST_ALPHA, ONE) for front-to-back composition. Actually, during frame-buffer writing, when alpha blending is enabled, the fragment value is mixed with the current pixel value according to the following equation:

$$C_{d_{new}} = C_s F_s + C_d F_d \tag{9.16}$$

where C_s is the source color coming from th fragment, C_d is the destination color of the pixel, F_s and F_d are, respectively the source and destination factor previously defined. Hence, if we want to express all components, the equation 9.16 becomes:

$$back-to-front:\begin{cases} R_{d_{new}} = R_s + R_d\left(1-\alpha_s\right) \\ G_{d_{new}} = G_s + G_d\left(1-\alpha_s\right) \\ B_{d_{new}} = B_s + B_d\left(1-\alpha_s\right) \\ \alpha_{d_{new}} = \alpha_s + \alpha_d\left(1-\alpha_s\right) \end{cases}$$

$$front-to-back:\begin{cases} R_{d_{new}} = R_s\left(1-\alpha_d\right) + R_d \\ G_{d_{new}} = G_s\left(1-\alpha_d\right) + G_d \\ B_{d_{new}} = B_s\left(1-\alpha_d\right) + B_d \\ \alpha_{d_{new}} = \alpha_s\left(1-\alpha_d\right) + \alpha_d \end{cases}.$$

In particular, the resulting opacity, after N iterations, is:

$$\begin{aligned} \alpha_N &= \alpha_n + \alpha_{N-1}\left(1-\alpha_n\right) \\ &= 1 - 1 + \alpha_n + \alpha_{N-1} - \alpha_{N-1}\alpha_n \\ &= 1 - \left(1-\alpha_n\right)\left(1-\alpha_{N-1}\right) \\ &= 1 - T_N \end{aligned} \tag{9.17}$$

when back-to-front accumulation is performed and:

$$\begin{aligned} \alpha_N &= \alpha_n\left(1-\alpha_{N-1}\right) + \alpha_{N-1} \\ &= 1 - 1 + \alpha_n - \alpha_n\alpha_{N-1} + \alpha_{N-1} \\ &= 1 - \left(1-\alpha_n\right)\left(1-\alpha_{N-1}\right) \\ &= 1 - T_N \end{aligned} \tag{9.18}$$

when front-to-back accumulation is performed, according to 9.3. Notably, equations 9.17 and 9.18 are valid only when associated color are used. But it is possible, by setting the blending factors to (SRC_ALPHA, ONE_MINUS_SRC_ALPHA), to perform back-to-front accumulation with pure colors, since $(R_{d_{new}}, G_{d_{new}}, B_{d_{new}})$ do not depend on destination opacity α_d, that, in this case, cannot be computed following the equation 9.3.

9.5.7. Multi-Texture Interpolation

In order to enhance the image quality of 2D texture based volume rendering, an approach to remove the visual artifacts caused by the fixed

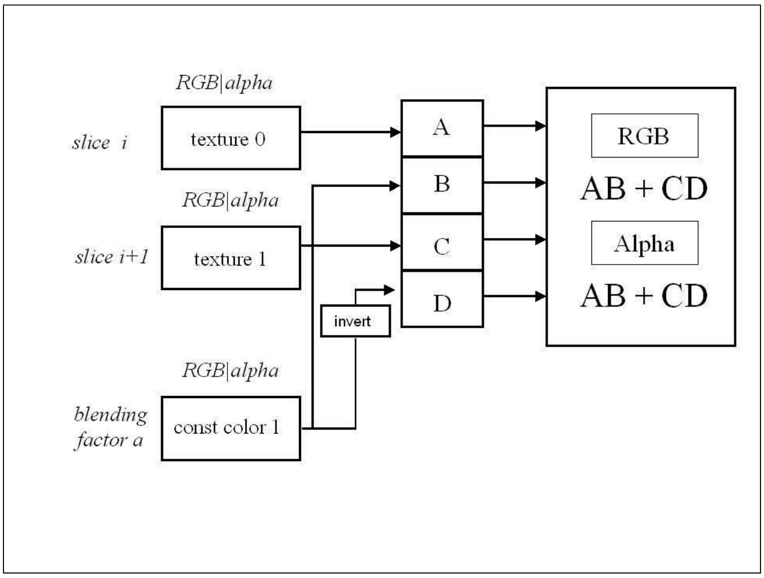


Figure 9.3.: **Multi-texture interpolation.** The combiner can be used to compute trilinear interpolation.

number of slices is required. We followed the multi-texture interpolation approach proposed by Ertl and others [WE98, RSEB$^+$00, EKE01], in which intermediate slices are computed on the fly in order to enable real trilinear interpolation. The third interpolation step is performed using NVidia OpenGL combiners extensions [Kil00]. The figure 9.3 displays the setup used to compute an intermediate slice $T_{i+\alpha}$ as a blending operation of two adjacent fixed slices T_i and T_{i+1} :

$$T_{i+\alpha} = (1 - \alpha) \cdot T_i + \alpha \cdot T_{i+1}$$

The fixed slices T_i and T_{i+1} are specified as **texture 0** and **texture 1** using the multi-texture extension. The combiner is setup to compute a component-wise weighted sum AB + CD with the interpolation factor α stored in one of the constant color registers. This application of multi texturing greatly enhances image quality by removing visual artifacts.

9.6. Performance enhancement

Pixel fill-rate is the major limiting factor when using a texturing approach to volume visualization. In zoom rendering, an appropriately down-scaled image is rendered in the back buffer and then enlarged and copied to the front buffer [MSG95]. In this way, delays associated with buffer swap synchronization are avoided, and the number of pixels filled during volume rendering is reduced. In our implementation, the copy and zoom operations are implemented by copying the reduced size image in texture memory and then rendering a textured polygon in the front buffer. Hence, sophisticated texture interpolation algorithms can be used to reduce the artifacts caused by magnification. Zoom rendering is particularly useful for medical volume rendering applications, because the pixel resolution is much larger than the resolution of the data that is displayed in the window. Zoom rendering is only active during navigation, still images being recomputed at maximum resolution [GPZT98].

9.7. Implementation and results

We tested our volume renderer on a standard PC equipped with 800 Mhz Intel Pentium IV CPU and a graphics board with NVidia Geforce 4 Ultra processor and 64 MB of DDRAM. Currently, with this hardware configuration, rendering times are about 70 msec. Figure 9.4 shows a snapshot of the volume visualization system during a virtual session of the surgical simulator.

9.8. Conclusion and discussion

This chapter discussed the volume renderer component that we designed and implemented for our temporal bone surgery training virtual system. We provide real–time visual feedback in parallel with the simulation of the physical system with a direct volume rendering approach. Rendering such a dynamic volume under real-time constraints is particularly challenging. In our approach, a fast approximation of the diffuse shading equation [Max95] is computed on the fly by the graphics pipe-line directly from the scalar data. We do this by exploiting the possibilities offered by multi-texturing with the register combiner OpenGL extension, that provides a configurable means to determine per-pixel fragment coloring [Kil01]. The extension is available on commodity graphics boards (e.g., NVIDIA GeForce series). Object-aligned volume slices are composed back-to-front. The Lambert shading equation is implemented in the graphics hardware by programming the register combiners, using multi-texturing to compute intermediate slices and approximate opacity gradients with forward differences. Gradient norms, that provide “surface strength” [DCH88b], are computed using a second order approximation of the square root programmed with the register combiners. This procedure is extremely efficient, since all the computation is performed in parallel in the graphics hardware and no particular synchronization is needed between the renderer and the process that is modifying the dataset. Only a single sweep through the volume is needed, and volume slices are sequentially loaded into texture memory on current standard PC graphics platform using AGP 4X transfers, which provide a peak bandwidth of 1054 MB/s.

9.9. Bibliographical notes

More informations about the rendering approaches described in this chapter can be found in [AGG$^+$02b, AGG$^+$03c]. Some of the contents of this chapter are also taken from reference [ZAG03].

Algorithm 1 Texture mapping code

```
// definisce la slice per l'unit di texture

  glActiveTextureARB( GL_TEXTURE0_ARB);
    glBindTexture (GL_TEXTURE_2D, tex(z) );

  glActiveTextureARB( GL_TEXTURE1_ARB);
    glBindTexture (GL_TEXTURE_2D, tex(z) );

  glActiveTextureARB( GL_TEXTURE3_ARB);
    glBindTexture (GL_TEXTURE_2D, tex(z) );

  glActiveTextureARB( GL_TEXTURE3_ARB);
    glBindTexture (GL_TEXTURE_2D, tex(z+dz) );


// Posiziona le slices sul poligono con gli offsets corretti.
// L'hardware fa il resto

  glMultiTexCoord2fARB(GL_TEXTURE0_ARB, smin, tmin);
  glMultiTexCoord2fARB(GL_TEXTURE1_ARB, smin+ds, tmin);
  glMultiTexCoord2fARB(GL_TEXTURE2_ARB, smin, tmin+dt);
  glMultiTexCoord2fARB(GL_TEXTURE3_ARB, smin, tmin);
    glVertex3f( xmin, ymin, zpos );

  glMultiTexCoord2fARB(GL_TEXTURE0_ARB, smax, tmin);
  glMultiTexCoord2fARB(GL_TEXTURE1_ARB, smax+ds , tmin);
  glMultiTexCoord2fARB(GL_TEXTURE2_ARB, smax, tmin+dt);
  glMultiTexCoord2fARB(GL_TEXTURE3_ARB, smax, tmin);
    glVertex3f( xmax, ymin, zpos);

  glMultiTexCoord2fARB(GL_TEXTURE0_ARB, smax, tmax);
  glMultiTexCoord2fARB(GL_TEXTURE1_ARB, smax+ds, tmax);
  glMultiTexCoord2fARB(GL_TEXTURE2_ARB, smax, tmax+dt);
  glMultiTexCoord2fARB(GL_TEXTURE3_ARB, smax, tmax);
    glVertex3f( xmax, ymax, zpos);

  glMultiTexCoord2fARB(GL_TEXTURE0_ARB, smin, tmax);
  glMultiTexCoord2fARB(GL_TEXTURE1_ARB, smin+ds, tmax);
  glMultiTexCoord2fARB(GL_TEXTURE2_ARB, smin, tmax+dt);
  glMultiTexCoord2fARB(GL_TEXTURE3_ARB, smin, tmax);
    glVertex3f( xmin, ymax, zpos);
```

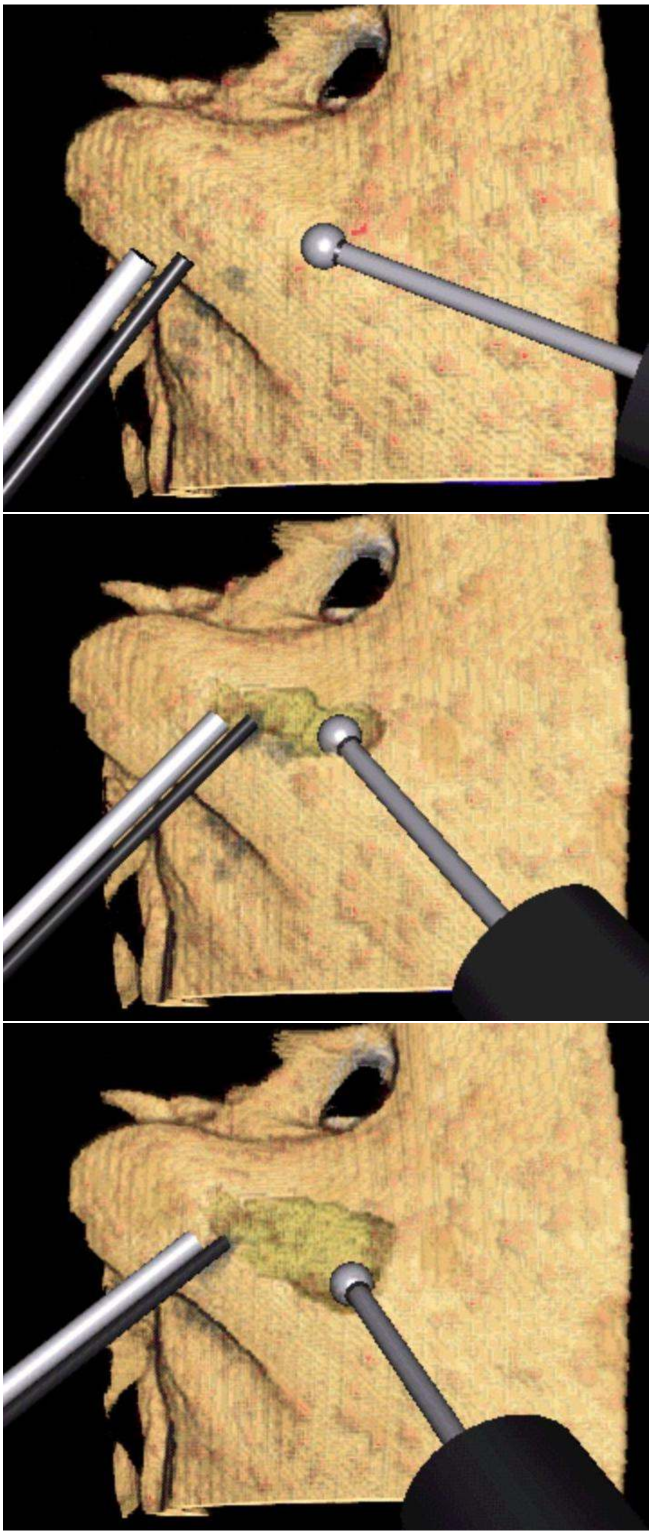
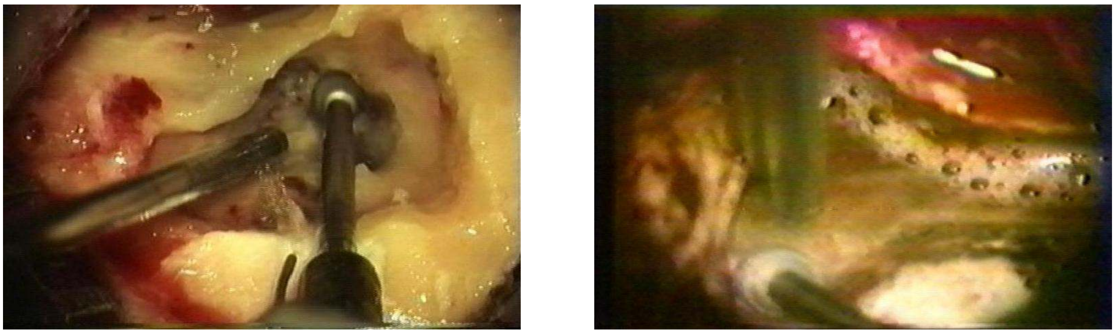


Figure 9.4.: **Snapshot of the volume.**

10. Secondary Effects Visual Simulation

This chapter describes the secondary effects visual simulation implemented as component of the surgical training system. The visual component uses a time-critical particle system evolution method to simulate secondary visual effects, such as bone debris accumulation, blooding, irrigation, and suction.

10.1. Introduction



(a) Mud (bone past) formation

(b) Obscuring effects

Figure 10.1.: **Operation scene.** These two images are typical examples of what is seen by the surgeon while performing mastoidectomy. In (a) it is clearly visible the paste created by the mixing of bone dust with water. If the paste and the water are not removed, they can obscure the field of view (b). Photos courtesy of Prof. Stefano Sellari Franceschini, Ear Nose and Throat Surgery, Dept. of Neuroscience, University of Pisa.

Obscuring effects are very important in a training system for temporal bone surgery. Actually, during bone dissection, the surgeon holds in his hands a high speed burr and a suction device, that he uses, respectively, to dissect the bone and to remove the bone paste produced by the mixing of bone dust with the water used to wash the burring region and to cool the burr bit. The capability of replicating the effects caused by the intertwining of the different physical processes is of primary importance for training. The absence of these effects would reduce, for instance, the importance placed by a trainee on the need for regular irrigation and suction [AGG^{+}02a]. In figure 10.1a the paste created by the mixing of bone dust with water is clearly visible, while figure 10.1b shows that if the past and water are not removed, they can obscure the field of view.

Although the presence of the water/paste mixture is essentially irrelevant with respect to the interaction between the burr and the bone, its presence cannot be neglected in the creation of the visual feed–back, since its “obscuring” effects constitute the principal cue to the user for the use of the sucker device. Therefore, we are modeling the dust/fluid dynamics using what essentially amounts to an hybrid particles/sand pile model [RS99, LM93]. The dust/fluid system is fed by the burring but its dynamics does not influence the haptic force evaluation.

10.2. Related work

A direct, “physically correct”, simulation of the dust-water system would require, to be able to capture all the dynamically relevant length scales, a very fine spatial resolution and it would be computationally incompatible with the real–time requirements of the simulation. For this reasons, secondary effects were mostly neglected in prior bone burring simulations. Actually, water and fluid simulation is a challenging task extensively studied by the scientific community. But currently methods are designed in order to obtain photorealism in the behavior of the simulated fluid surfaces [EMF02], and they are far from achieving real-time performances. The simulation of complex water effects using the full 3D Navier-Stokes equations has been based upon the large amount of research done by the computational fluid dynamics community over the past 50 years. Foster and Metaxas [FM96] first developed a 3D Navier-Stokes methodology for the realistic animation of liquids. Further CFD enhancements to the traditional marker and cell method which allow one to place particles only near the surface can be found in [CdVL95]. A semi-Lagrangian “stable fluids” treatment of the convection portion of the Navier- Stokes equations was introduced to the computer graphics community by Stam [Sta99] in order to allow the use of significantly larger time steps without hindering stability. Foster and Fedkiw [FF01] made significant contributions to the simulation and control of three dimensional fluid simulations through the introduction of a hybrid liquid volume model combining implicit surfaces and massless marker particles; the formulation of plausible boundary conditions for moving objects in a liquid; the use of an efficient iterative method to solve for the pressure; and a time step sub-cycling scheme for the particle and implicit surface evolution equations in order to reduce the amount of visual error inherent to the large semi-Lagrangian “stable fluid” time step used for time evolving the fluid velocity and the pressure. The combination of all of the above advances in 3D fluid simulation technology along with ever increasing computational resources has set the stage for the inclusion of fully 3D fluid animation tools in a production environment. Relatively the dust generation, he most relent work was recently made by Chen et al [CFW99] for real-time simulation of dust behavior generated by a traveling vehicle. They use particle systems, computational fluid dynamics, and behavioral simulation techniques to simulate dust behavior in real time, by analyzing the forces and affecting factors, and by constructing physically-based empirical models to generate dust particles. Prior surgical simulation systems did not comprise water and dust secondary visual effects. The Ohio Virtual Temporal Bone Dissection simulator simply removes voxels by making them transparent [WBS+00, BSWS01]. Localized bleeding is simulated by coloring in red the voxels close to the burr bit. Our system [AGG+03b] exploits the difference in frequency requirements of the visual and haptic simulations by running a rule-based particle system simulator in parallel with the bone dissection simulator. The method is able to provide a crude visual approximation of bone debris accumulation, blooding, irrigation, and suction. Furthermore, we improve the technique by introducing a time-critical particle evolution method that trades simulation quality with time.

10.3. Bone dust, debris and water simulation

As it was mentioned before, the burring of the bone produces dust that mixes with water in a paste, “mud”, clearly visible in figure 10.1(a). The paste material has a quite complex behavior, from sand–like to gel–like. The water paste mixture needs to be continuously removed, otherwise it can obscure the field of view, as shown in figure 10.1(b).

Given these required behaviors, the physical system that we want to represent in our virtual simulator is composed by the following

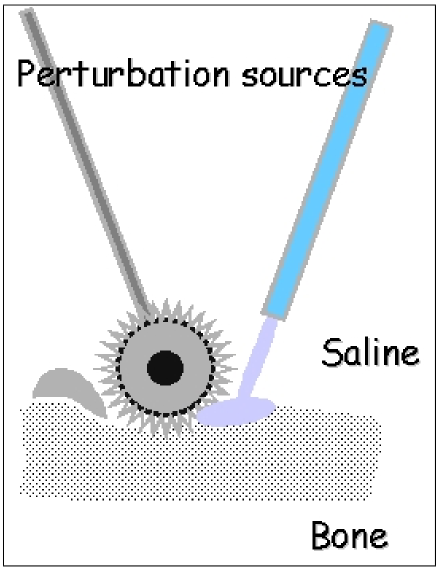


Figure 10.2.: **Physical system**: it can be considered as composed by externally controlled perturbation sources and by a global scene.

items (see figure 10.2):

- externally controlled perturbation sources, i.e. the burr and the sucker; their effects on the system can be considered as locally adding or removing of conserved quantities. These physical effects depend on tools position and intensity, that are user-controlled quantities.
- global scene, that is composed by the operational site plus the water injected by the sucker.

The evolution of the global scene may be studied in a lot of ways, with any kind of assumption or simplification. We are modeling the dust/fluid dynamics using a hybrid particles-volumetric model, inspired by previous work on particle systems and sand-piles [RS99, LM93].

Each particle has a mass, a position, a velocity and a dynamic behavior. Water particles are introduced by the irrigator with an initial velocity directed along the irrigator axis. Dust particles are generated by the burr performing the surgical bone drilling with an initial velocity depending on the rotation of the burr itself and a creation rate depending on the mass flux. Blood particles are generated by tissues with negligible initial speed. All particles move according to Newton's law when free, and interact with the other materials according to a set of rules that ensure that only a single particle may occupy a given

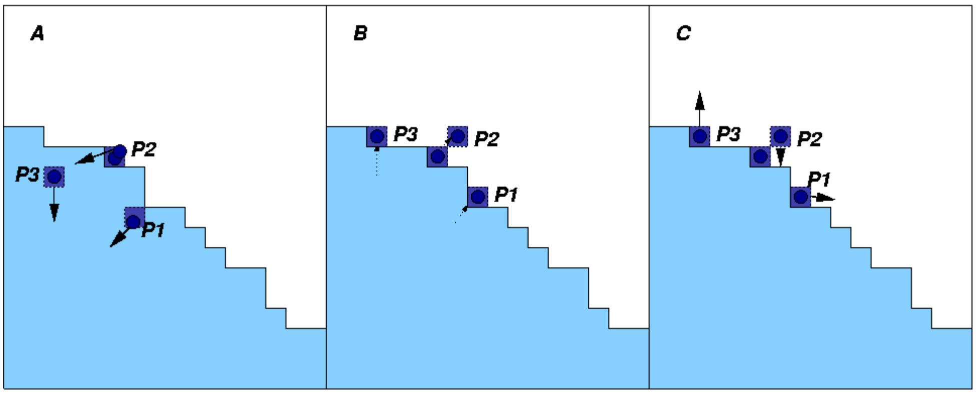


Figure 10.3.: **Particle behavior**: particles interact with the other materials according to a set of rules that ensure that only a single particle may occupy a given voxel at given time.

voxel at given time. Basically, when a particle enters a non empty voxel, it is reflected back-wards to the first free voxel (see fig. 10.3). Its state is then modified as a function of the colliding materials and the particle velocity. When a particle collides with the environment, we choose between elastic scattering or sliding along the bone surface based on the particle velocity. The random choice is made according to a probability distribution that favors scattering for high impact velocities. Different materials are modeled by shaping the probability distribution and by defining different particles masses and reflection coefficients. In particular, bone particles have a behavior similar to water, but higher mass and higher probability to be scattered by hard bone.

We model bone paste formation by changing the material of bone and water particles to “bone paste” when they collide. We also consider the interaction of particles with the burr, by scattering away the particles that enter in contact with the burr bit with a velocity depending on the rotational axis and speed of the burr.

10.4. Adaptive technique

Although the presence of the water/paste mixture is essentially irrelevant with respect to the interaction between the burr and the bone, its presence cannot be neglected in the creation of the visual feed–back, because its “obscuring” effects constitute the principal cue to the user for the use of the suction device[AGG$^+$02a].

For the computational reasons discussed in section 7.2, we are modeling the dust/fluid dynamics using what essentially amounts to an hybrid particles-volumetric model, inspired by previous work on particle systems and sand-piles [RS99, LM93]. In this scheme, particles are created by the irrigator, which injects water particles, by blood spots and vessels, that inject blood particles, and by the burr during erosion, that converts bone to bone dust particles. All particles move ballistically when in empty space, and interact with the other materials according to a set of rules that ensure that only a single particle may occupy a given voxel at given time (see [AGG+02b]). Particles are deleted when they exit from the operation site or when they are sucked by the suction device.

The computational cost of update in this scheme is essentially constant per particle and, thus, the total computation cost would naively grow linearly in the number of particles and quickly degrade the real-time performance of the system. To avoid this problem, we are using a time-critical evolution algorithm designed to trade simulation quality with speed. The idea behind the algorithm is to concentrate resources on the visually most important parts of the simulation, by controlling both individual particles update rates and total number of particles.

The update rate control methods associates to each particle an update rate proportional to the particle speed. To avoid the costs associated to sorting the particles, the particles are divided in groups, $\{G_i\}$, so that all the particles in group G_i have speed v, measured in units of a predefined reference maximal velocity scale, in the range $2^{-i} \leq v < 2^{-(i+1)}$. Particle velocities are clamped so that they cannot be larger that the maximal velocity scale. At each evolution time step we randomly select $\{a_i\}$ particles from each group and, for each selected particle, integrate the motion from its last recorded time of update to the current time.

The effective time step for particles in group G_i is then $(dt)_i = n_i/a_i(dt)_\mu$ where n_i is the number of particles in group G_i and $(dt)_\mu$ is the actual simulation time step. The selection counters $\{a_i\}$ are chosen so that, on average, particles in channel i will move with a time step $(dt)_i = 2(dt)_{i-1}$, and thus

$$\frac{a_{i+1}}{n_{i+1}} = \frac{1}{2}\frac{a_i}{n_i}. \tag{10.1}$$

The total computational cost for one time step will then be $W = w \sum_i a_i$ where w is the average cost per particle update, which is measured at run time by the simulator. Using the equation above we find that, when all the $n_i > 0$,

$$A = \sum_i a_i = \frac{a_0}{n_0} \sum_i \frac{n_i}{2^i}. \tag{10.2}$$

Therefore, for given W, w, and n_i, we can reconstruct the required a_i. The case $n_j = 0$ for some j is a trivial generalization of the above.

Given a reasonable approximation of w, the update rate control algorithm is guaranteed to meet timing constraints and to probabilistically move the particles with the largest visual error. If the update rate of the particle system falls below a specified threshold (currently, if we move less that 10% of the particles per step), we reduce the particle count by removing the “less important” ones. The importance of a particle is currently inversely proportional to the distance from the current look-at point of the microscope and to the particle velocity.

10.5. Conclusion and Discussion

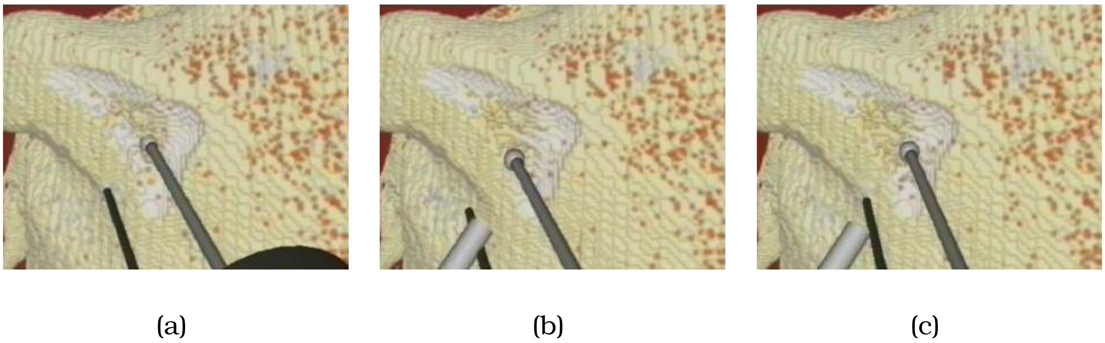

(a) (b) (c)

Figure 10.4.: **A virtual burring sequence.** Here we show a typical bone cutting sequence performed in the mastoid region. The accumulation of debris, and its masking effects, is clearly visible.

A direct, “physically correct”, simulation of the dust-water system would require, to be able to capture all the dynamically relevant length scales, a very fine spatial resolution and it would be computationally incompatible with the real–time requirements of the simulation. For this reasons, secondary effects were mostly neglected in prior bone burring simulations. Our system [AGG^{+}02a] exploits the difference in frequency requirements of the visual and haptic simulations by running a rule-based particle system simulator in parallel with the bone dissection simulator. The method is able to provide a crude visual approximation of bone debris accumulation, blooding, irrigation, and suction. Fig. 10.4 shows a typical erosion sequence. The accumulation of debris, and its masking effects, is clearly visible. In order to improve the realism of the dust/fluid simulation, we plan to introduce in the next version of the simulator a more sophisticated treatment of particle collisions and thus to remove limitations such as the single particle per voxel constraint of the current implementation.

10.6. Bibliographical notes

The subjects described in this chapter were taken from references [AGG$^+$03b, AGG$^+$02b].

Part IV.

Implementation and Results

The results of the human factors analysis indicate that, to be able to feed the appropriate sensorial inputs to the human perceptual system, the system needs to produce data at two very different time-scales: about 15-20 Hz for the visual rendering, and around 1 KHz for the haptic response. We have exploited this difference in complexity and frequency requirements by modeling the simulator as a collection of loosely coupled concurrent components. The system is based on patient-specific volumetric object models derived from 3D CT and MR imaging data. Real-time feedback is provided to the trainees via direct volume rendering and haptic feedback. The performance constraints dictated by the human perceptual system are met by exploiting parallelism via a decoupled simulation approach on a multi-processor PC platform. This part of the dissertation outlines the main components of the system, as implemented in our current prototype, and preliminary results obtained during training sessions. Specifically, chapter 11 describes the training system current configuration, chapter 12 provides preliminary results reguarding the end-user evaluation of the simulator, while chapter 13 discusses preliminary results obtained during psychophysical testing sessions.

11. System Integration

In this chapter, the components of the training system for simulating temporal bone surgery are presented and detailed. The system is based on patient-specific volumetric object models derived from 3D CT and MR imaging data. Real-time feedback is provided to the trainees via real-time volume rendering and haptic interfaces. The performance constraints dictated by the human perceptual system are met by exploiting parallelism via a decoupled simulation approach on a multi-processor PC platform.

11.1. Introduction

In this chapter we describe the design choices that we followed in the development of the training system for simulating surgery on the temporal bone, a skull region just behind the ear. The ability to rehearse the procedure using patient specific data is extremely rare, so a VR simulator realistically mimicking a patient-specific operating environment should address this shortcoming. We designed a training system in order to work on patient-specific volumetric object models directly derived from 3D CT and MRI images, and to provide realistic visual and haptic feedback, including secondary effects such as the obscuring of the operational site due to the accumulation of bone dust and other burring debris. It is expected that the ability of using directly patient specific data as input will help in the accumulation of a large number of training cases. Moreover, it will open the road toward the use of the simulator for pre-operation planning and rehearsal, making thus possible to plan surgery directly on a model of the individual patient, rather than by referring to a model surgical procedure on a standard anatomy. The need to provide real time feedback to users, while simulating burring and related secondary effects, imposes stringent constraints on the system. Our solution is based on a volumetric representation of the scene, and it harnesses the locality of the physical system evolution to model the system as a collection of loosely coupled components running in parallel on a multi-processor PC platform. This chapter focuses on the general system architecture. The rest of the chapter is structured as follows. In section 11.2 we describe the implied real time constraints and how they can be satisfied with a decoupled simulation model. Section 11.3 describes the current prototype system architecture, while section 11.4 reports on current hardware system configuration.

11.2. The Decoupled Simulation Model

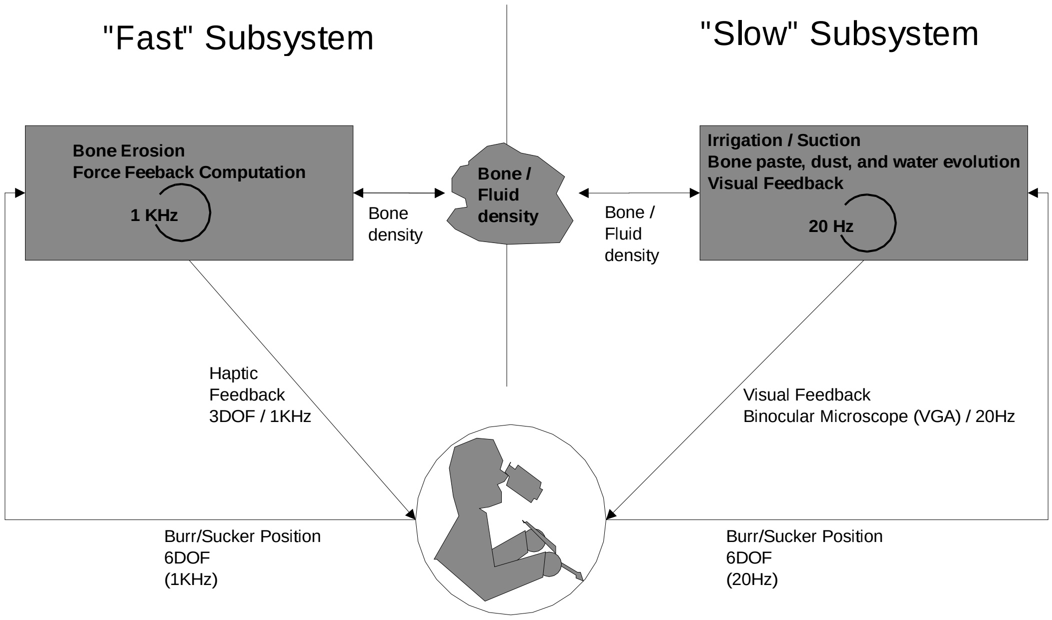

"Fast" Subsystem
"Slow" Subsystem
Bone Erosion
Force Feeback Computation
1 KHz
Bone density
Bone / Fluid density
Bone / Fluid density
Irrigation / Suction
Bone paste, dust, and water evolution
Visual Feedback
20 Hz
Haptic Feedback 3DOF / 1KHz
Visual Feedback Binocular Microscope (VGA) / 20Hz
Burr/Sucker Position 6DOF (1KHz)
Burr/Sucker Position 6DOF (20Hz)

Figure 11.1.: **Logical system decomposition.** The system is divided in a ”fast” subsystem, responsible for the high frequency tasks (surgical instrument tracking, force feedback computation, bone erosion), and a ”slow” subsystem, essentially dedicated to the production of data for visual feedback.

The results of the human factors analysis (see chapter 4) indicate that, to be able to feed the appropriate sensorial inputs to the human perceptual system, the system needs to produce data at two very different time-scales: about 15-20 Hz for the visual rendering, and around 1 KHz for the haptic response [AGG$^+$02b]. The computations needed to obtain the haptic force response can be drastically simplified, since response forces can be computed by just considering a small neighborhood around the contact surfaces between surgical instruments and bones. The simulation of secondary effects and the visualization of the evolving operating theater requires, however, a larger computational effort. We have exploited this difference in complexity and frequency requirements by modeling the simulator as a collection of loosely coupled concurrent components. Logically, the system is divided in a ”fast” subsystem, responsible for the high frequency tasks (surgical instrument tracking, force feedback computation, bone erosion), and a ”slow” one, essentially dedicated to the production of data for visual feedback (see figure 11.1). The “slow” subsystem is responsible for the global evolution of the water, bone dust and bone paste.

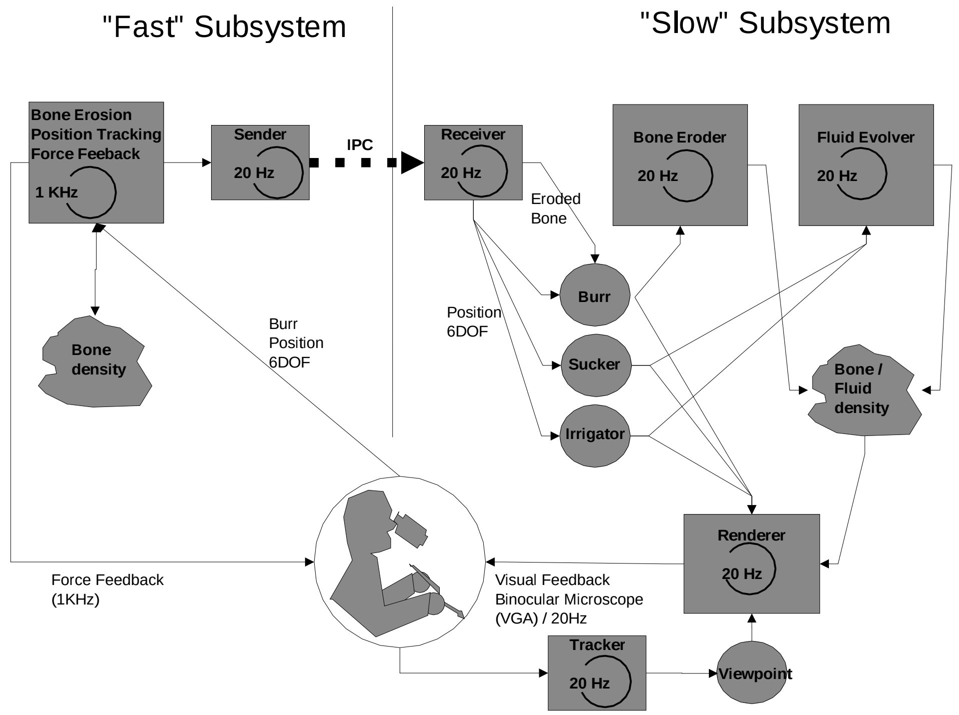


Figure 11.2.: **Decoupled simulation architecture.** The first machine is dedicated to the high-frequency tasks: haptic device handling and bone removal simulation. The second machine concurrently runs at 10-20 Hz the low-frequency tasks: bone removal, fluid evolution and visual feedback.

These secondary effects can be considered purely visual, since they just contribute to visual clutter without producing important forces to be returned to the user. The algorithms used to control the simulations are local in character and they are structured so that they communicate only via changes in the relevant, local, substance densities. This arrangement leads naturally to a further break-up of the slow subsystem in components, each dedicated to the generation of a specific visual effect, and thus to a parallel implementation on a multiprocessor architecture.

11.3. Prototype System Architecture

We have built a prototype system, simulating the effects identified by the task analysis. The system is running on a dual PC platform. It exploits both message passing and shared-memory parallelism to meet the performance constraints imposed by the human perceptual system. The system uses a volumetric approach, with the initial configuration of the model directly derived from patient CT data. The data is initially replicated on the two machines. The first machine is dedicated to the high-frequency tasks: haptic device handling, one for the dominant hand controlling the burr and the irrigator, and the other controlling the sucker, as well as bone removal simulation. These tasks require at least 1 kHz update frequency because of the need of simulating hard contacts. The second PC concurrently runs at 10-20 Hz the low-frequency tasks: bone dust evolution and visual feedback. The two machines are synchronized using one-way message passing with a dead reckoning protocol. In our volumetric description of the scene, voxels labeled as bone must bone must react to the manipulators through the haptic feedback devices, but they do not evolve unless they are removed by burring. In the data replicated in the machine dedicated to low frequency tasks, further values are introduced in the volume labeling voxels occupied by dust, blood and water. These values are used directly by the volume rendering thread. The ”fast” subsystem performs the burring simulation, i.e. the force feedback calculation and the bone removal from the dataset, sending the force value to the haptic devices, and sending information on manipulator positions and bone removed to the ”slow” subsystem. This task is extremely difficult to perform at over 1 Khz. We have thus organized our simulation so that each time step is divided into two sub-steps. The first sub-step estimates the bone material deformation and the resulting elastic forces, given the relative position of the burr with respect to the bone. The second sub-step estimates the local rate of cutting of the bone by using a postulated energy balance between the mechanical work performed by the burr motor and the energy needed to cut the bone, which is assumed to be proportional to the bone mass removed. The reader should refer to chapters 6 and 7 for major information on haptic simulation. The ”flow” subsystem performs the visual simulation of bone dust and fluid dynamics as well as the visual rendering of the scene. We are modeling the dust/fluid dynamics using what essentially amounts to a hybrid particles/sand pile model (see chapter 10). The visual rendering subsystem must operate within the timing constraints imposed by the human perceptual system (i.e. a latency of less than 300 ms, and a frequency above 10-15 Hz). We reach this goal by using a parallel processing approach, which exploits the capabilities of current graphics PC architectures. In our system, the renderer is totally decoupled from the simulator and the tracking system, and runs at its own frequency. The rendering system is based on a volumetric approach. We use texture mapping and alpha blending for a back to front reconstruction of the scene. Shading effects are implemented by exploiting the register combiner OpenGL extension on most modern commodity graphic boards. The rendering system is extensively described in chapter 9. Surgical instruments are rendered as polygons, and combined with the volumetric rendering of the rest of the scene using Z-buffering.

11.4. Hardware System Configuration

A prototype system, based on the techniques discussed above, is running on a dual PC platform. Our current configuration is the following:

- a single-processor PIV/1.5 GHz with 256 MB PC133 RAM for the high-frequency tasks; two threads run in parallel: one for the haptic loop (1KHz), and one for sending volume and instrument position updates to the other machine;
- a dual-processor PIII/800 MHz with 512 MB PC800 RAM and a NVIDIA GeForce 4 Ultra running a Linux 2.4 kernel for the low frequency tasks; three threads are continuously running on this machine: one to receive volume and position updates, one to simulate bone removal and fluid evolution, and one for visual rendering;
- a Phantom Desktop haptic device for the dominant hand; the device is connected to the single processor PC. It provides 6DOF tracking and 3DOF force feedback for the burr/irrigator;
- a Phantom 1.0 haptic device for the non-dominant hand; the device is connected to the single processor PC. It provides 6DOF tracking and 3DOF force feedback for the sucker;
- an n-vision VB30 binocular display for presenting images to the user; the binoculars are connected to the S-VGA output of the dual processor PC.

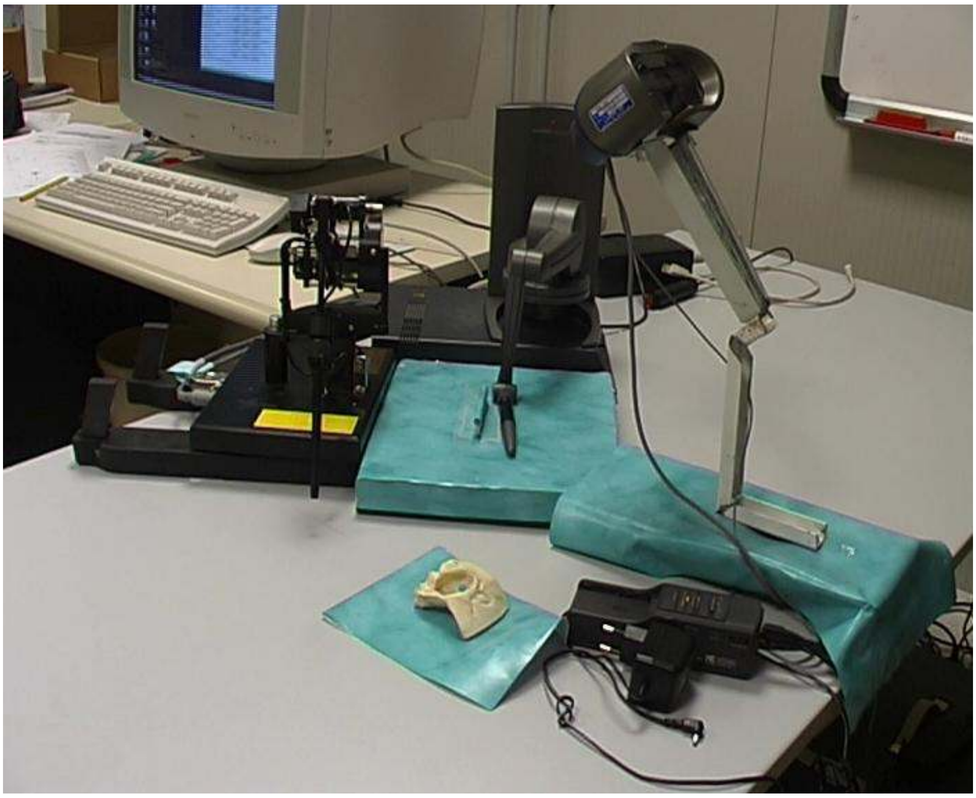


Figure 11.3.: **The current Ierapsi surgical simulator set-up**. Note the Phantom Desktop haptic device for the dominant hand; the Phantom 1.0 haptic device for the non-dominant hand and the n-vision VB30 binocular display.

11.5. Conclusion

We designed a training system in order to work on patient-specific volumetric object models directly derived from 3D CT and MRI images, and to provide realistic visual and haptic feedback, including secondary effects such as the obscuring of the operational site due to the accumulation of bone dust and other burring debris. The system is divided in a ”fast” subsystem, responsible for the high frequency tasks (surgical instrument tracking, force feedback computation, bone erosion), and a ”slow” subsystem, essentially dedicated to the production of data for visual feedback. The system uses a volumetric approach, with the initial configuration of the model directly derived from patient CT data. The data is initially replicated on the two machines. The first machine is dedicated to the high-frequency tasks: haptic device handling and bone removal simulation. The second machine concurrently runs at 10-20 Hz the low-frequency tasks: bone removal, fluid evolution and visual feedback. The two machines are synchronized using one-way message passing with a dead reckoning protocol.

11.6. Bibliographical notes

The contents of this chapter review references [AGG$^+$02b, AGG$^+$02a], where we described the details about the system design and architecture of our simulator for temporal bone surgery training.

12. End–User Testing And Validation

In this chapter, we describe the results of tests performed on our virtual surgical training system for middle ear surgery. The work aims to demonstrate how expert surgeons and trainees can effectively use the system for training and assessment purposes. Preliminary kinematic and dynamic analysis of simulated mastoidectomy sessions are presented.

12.1. Introduction

The data acquisition and analysis involves all the bone-burr interaction dynamic parameters in a series of simulated specific interventions performed by trainees and experienced surgeons. The specialty considered in these sessions is the basic mastoidectomy, that represents the most superficial and common surgery of the temporal bone, and it is undertaken by a wide range of surgeons in everyday practice. The procedure consists in the removal of the air cavities just under the skin behind the ear itself, and it is performed for chronic infection of the mastoid air cells (mastoiditis). The rest of the chapter is organized as follows. Section 12.2 provides a short description of the virtual surgical training system, while section 12.3 illustrates our preliminary results with regards to the surgical simulator testing as well as the kinematic and dynamic analysis of the basic mastoidectomy phases. Finally, section 12.4 provides some statistics analysis about the training sessions.

12.2. Methods and tools

We have gathered initial feedback about the prototype system from specialist surgeons from the University of Pisa that are collaborating to this research. Subjective input has been used to tune the parameters that control force feedback, as well as to compare experimental

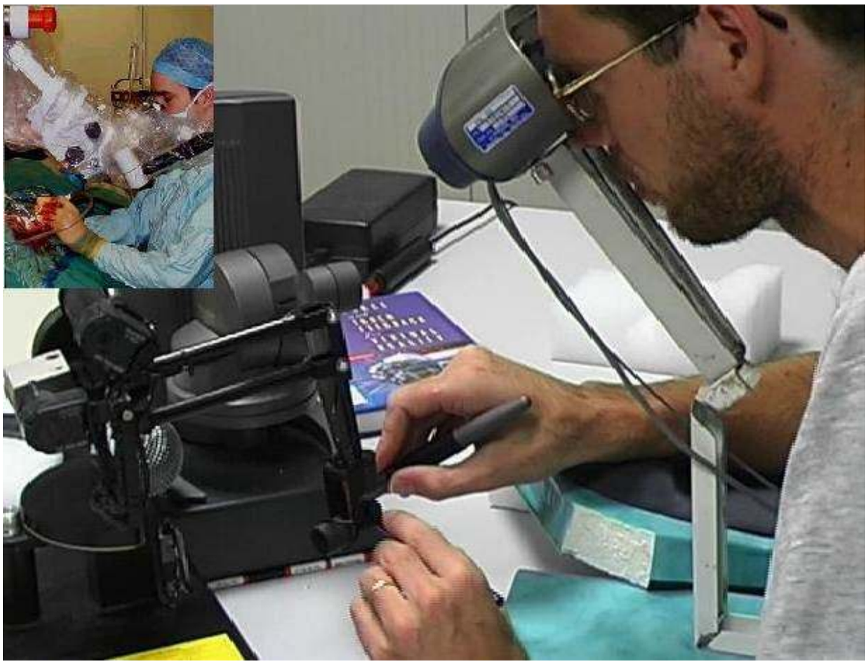


Figure 12.1.: **Virtual vs real mastoidectomy surgical setup**. The surgeon looks at the region interested by the procedure via a stereoscopic microscope and holds in his hands a high speed burr and a sucker. The main photo shows the virtual simulator setup, while the inset photo represents a typical mastoidectomy surgical theater. Inset photo courtesy of Prof. Bob Stone, Virtual Presence Ltd.

results. The overall realism of the simulation is considered sufficient for training purposes. Demonstration movies are available on CRS4 web site [1].

12.3. Results

We are extensively testing the virtual surgical training system in collaboration with surgeons of the Department of NeuroScience of the University of Pisa. In particular, contact model parameters and erosion factors have been tuned according to their indications and there is consensus that they represent a good approximation of reality. Using the tuned system, surgeons can perform complete virtual surgery procedures with satisfactory realism. Figures 12.2 and 12.3 show the comparison between a real mastoidectomy and a virtual one, performed on our surgical simulation system. The possibility of recording dynamic values of a surgical training session provides new opportunities for the analysis and the evaluation of procedures. Different surgical procedure could be recognized by the system and it becomes possible to use the recorded values also to compare the behavior of expert surgeons and trainees in order to evaluate surgical skills. Some interesting considerations can be done about the force and velocity histograms resulting from virtual mastoidectomy sessions. Specifically, figure 12.4 and 12.5 are histograms derived from virtual interventions performed by two trainees of University of Pisa, while figure 12.6 represents the histograms of an ENT expert surgeon forces and velocities (courtesy of Prof. Stefano Sellari Franceschini from University of Pisa). Current available data show consistency between different training sessions of the same user. Average forces exerted by burr are between 0.7 and 1.3 N for the expert surgeon and between 0.8 and 1.1 N for trainees, while average tool velocities are between 8.0 and 12.0 m/sec for the expert surgeon and 10.0 and 17.0 m/sec for trainees.

12.4. Mastoidectomy analysis

In order to evaluate the possibility of characterizing different procedures according to dynamical parameters computed by the simulator, we recorded all the parameters (i.e. burr and sucker positions and velocities, force vectors, voxels removed) during a series of simulated mastoidectomy procedures.We analyzed four steps of the mastoidectomy procedure. In the first, the surgeon removes the cortex. The drill

[1] www.crs4.it/vic/multimedia

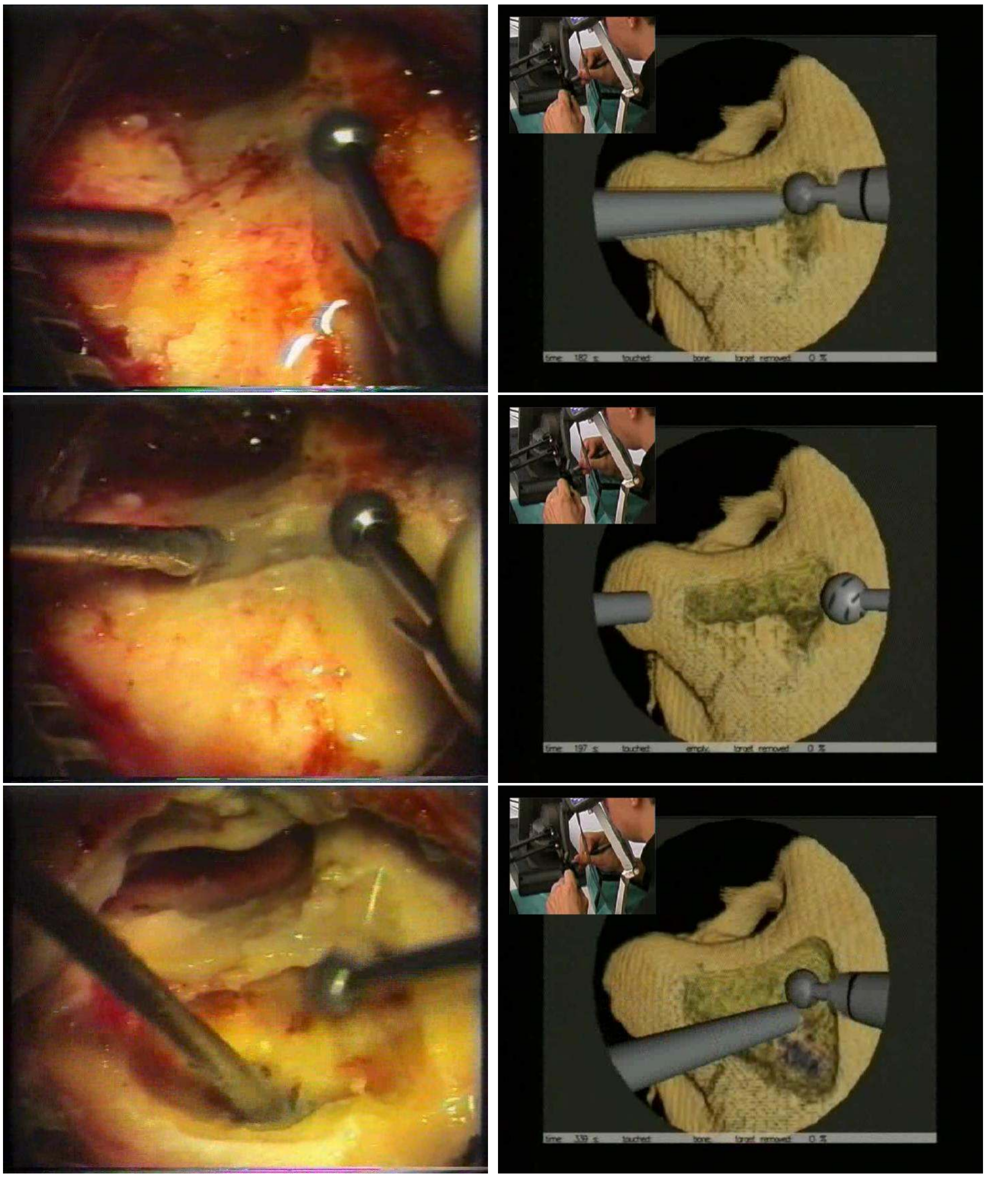

time: 182 s touched: bone: target removed: 0 %
time: 197 s touched: empty: target removed: 0 %
time: 339 s touched: bone: target removed: 0 %

Figure 12.2.: **Comparison between real and virtual intervention**: the principal steps of a basic mastoidectomy, performed by a surgeon are represented. Photos courtesy of Prof. Stefano Sellari Franceschini, Univerisity of Pisa.

is applied to the mastoid cortex immediately posterior to the spine of Henle and draws two perpendicular cuts, the first along the temporal line and the second toward the mastoid tip. Then the mastoid cortex is then removed in a systematic fashion of saucerization.

Figure12.7A shows a snapshot of the scene viewed by the trainee during this step and on the right plots of the force module and of the material removed as a function of time. The second step is the

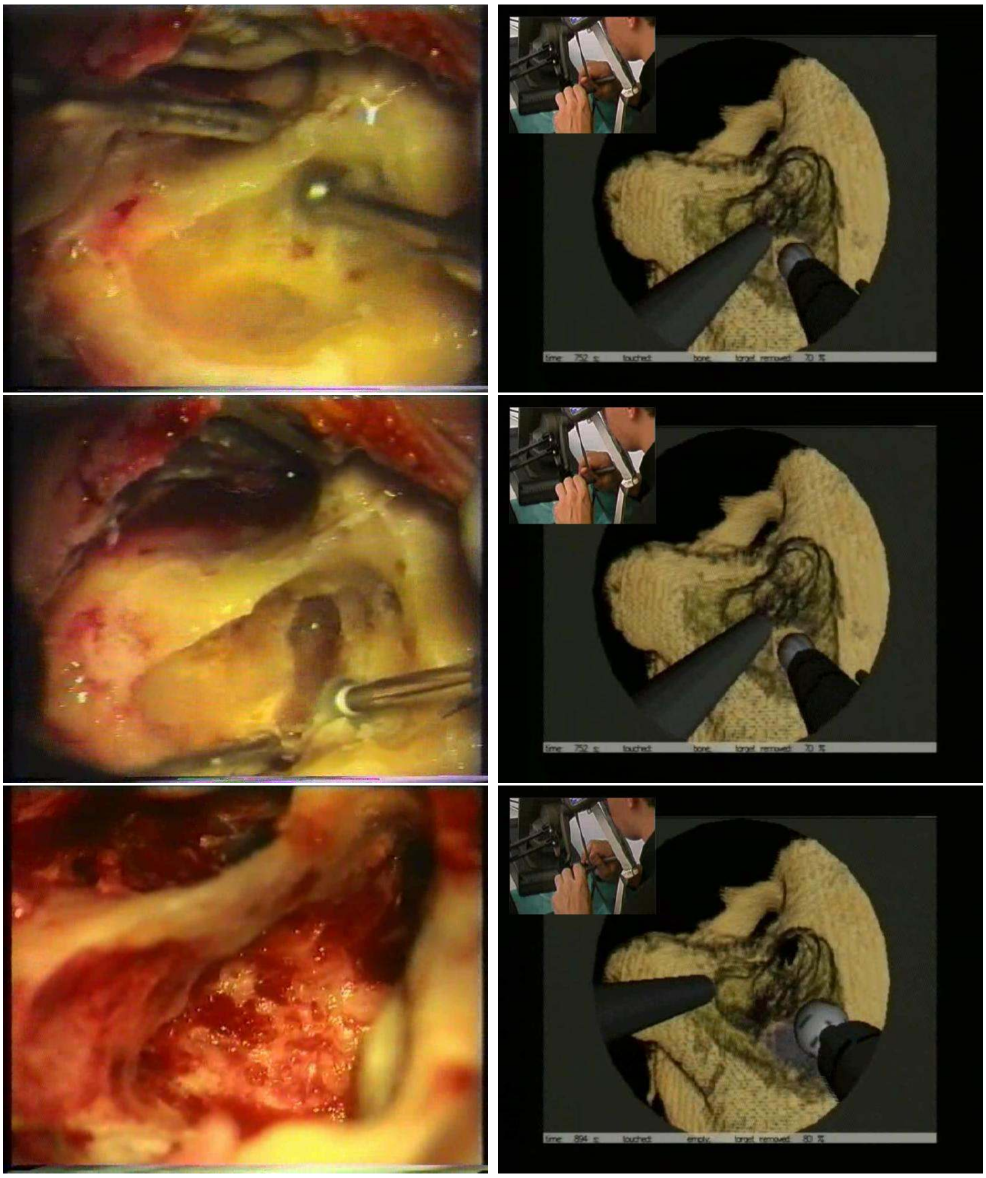

time: 752 s touched: bone: target removed: 70 %
time: 752 s touched: bone: target removed: 70 %
time: 894 s touched: empty: target removed: 80 %

Figure 12.3.: **Comparison between real and virtual intervention**: the principal steps of a basic mastoidectomy, performed by a surgeon are represented. Photos courtesy of Prof. Stefano Sellari Franceschini, Univerisity of Pisa.

cavity saucerization: before a deeper penetration in the antrum, it is necessary to perform a wide cortical removal and the posterior canal should be thinned so that the shadow of an instrument can be seen through the bone when the canal skin is elevated. Snapshot and plots relative to this step are shown in Figure12.7B. In the next phase considered there is the identification of the mastoid antrum. It can be identified as a larger air-containing space at whose bottom

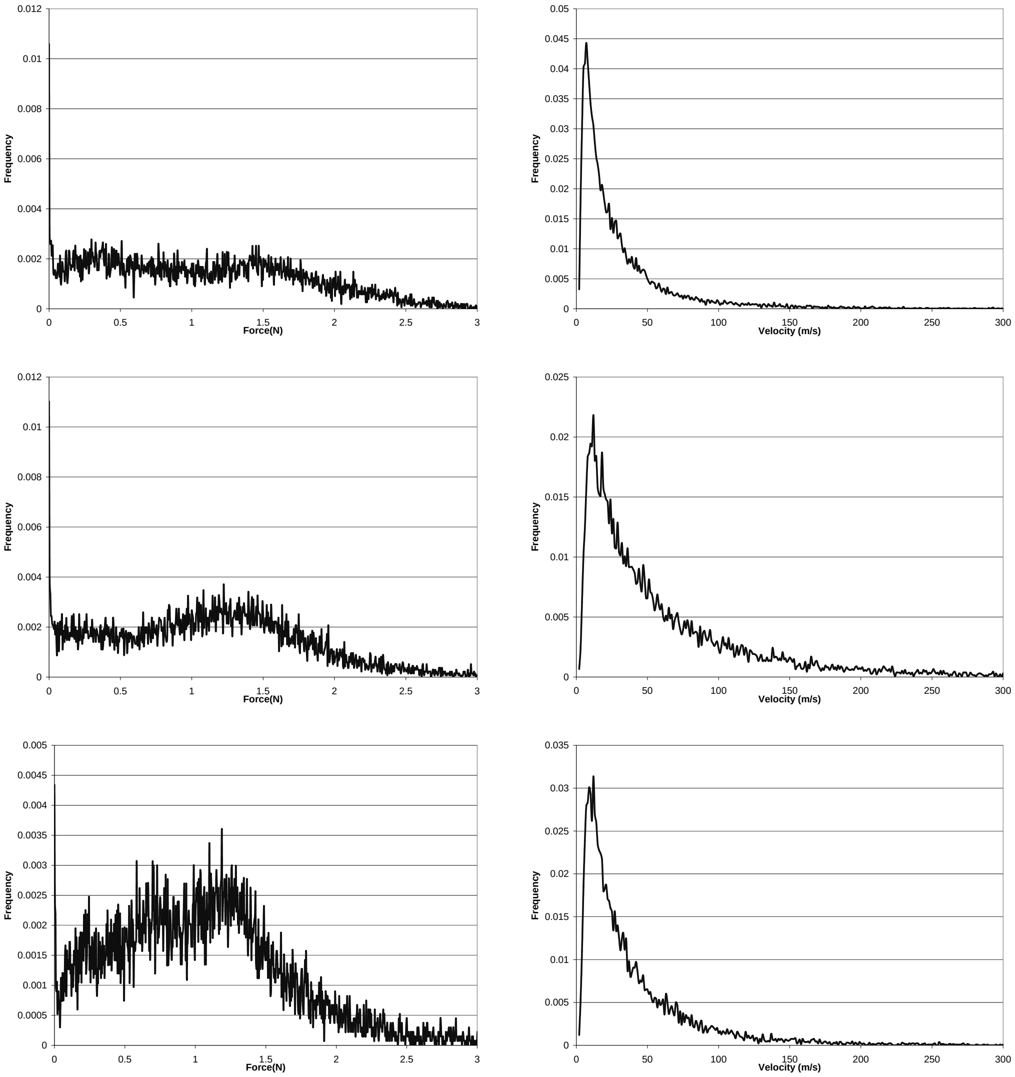


Figure 12.4.: **First training surgeon histograms**: force and velocity histograms resulting from surgical training system sessions.

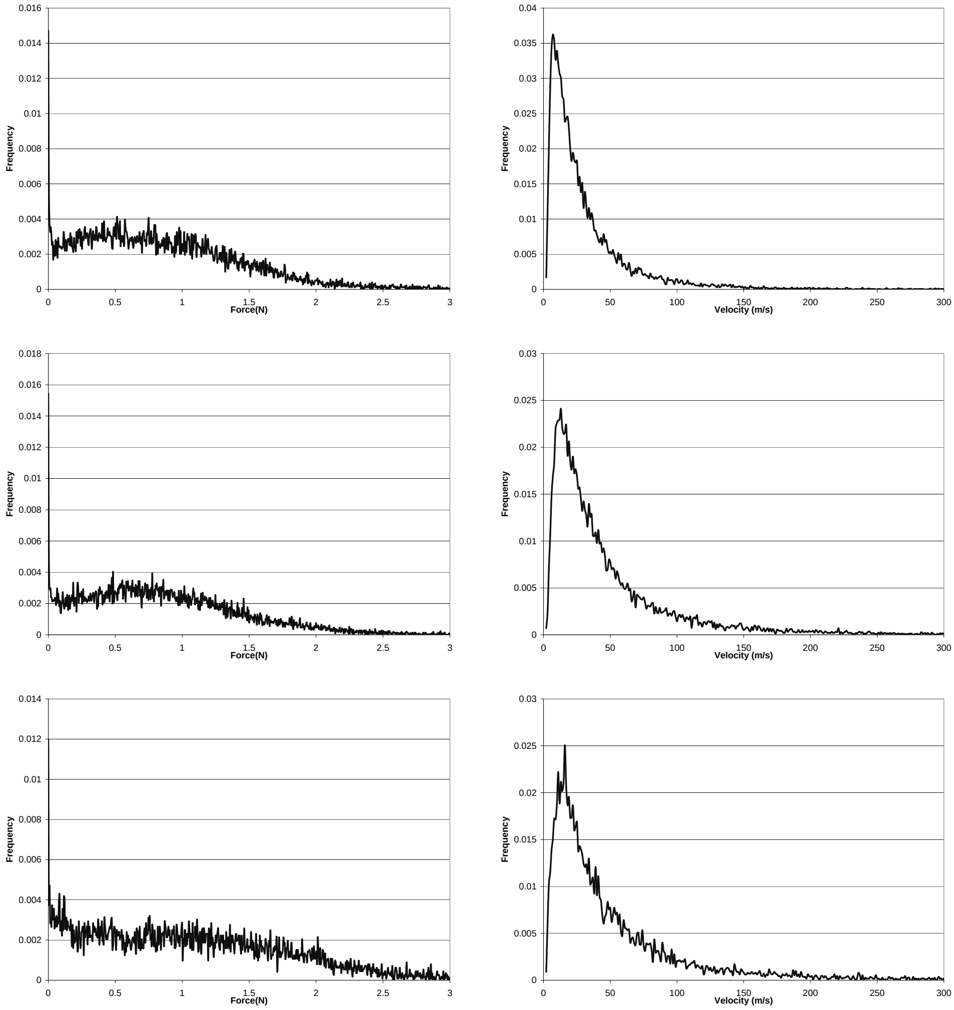


Figure 12.5.: **Second training surgeon histograms**: force and velocity histograms resulting from surgical training system sessions.

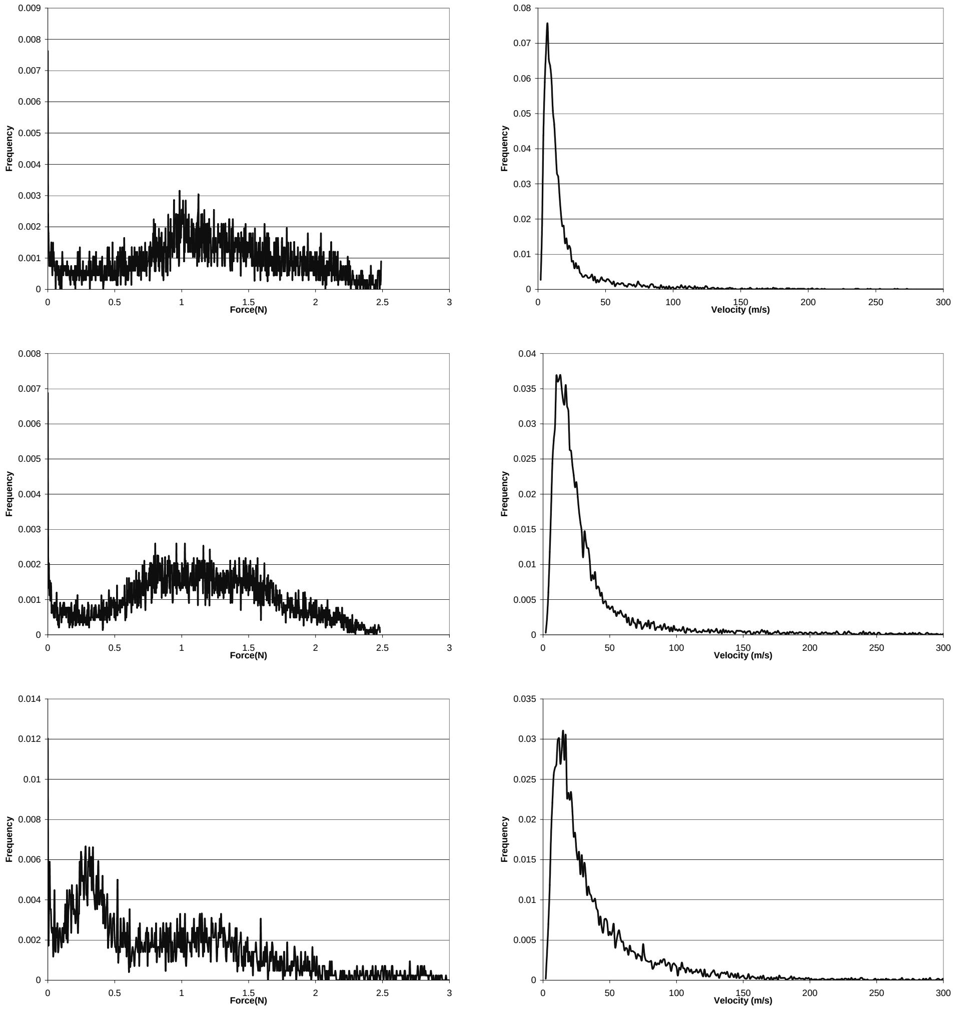


Figure 12.6.: **Expert ENT surgeon histograms**: force and velocity histograms resulting from surgical training system sessions.

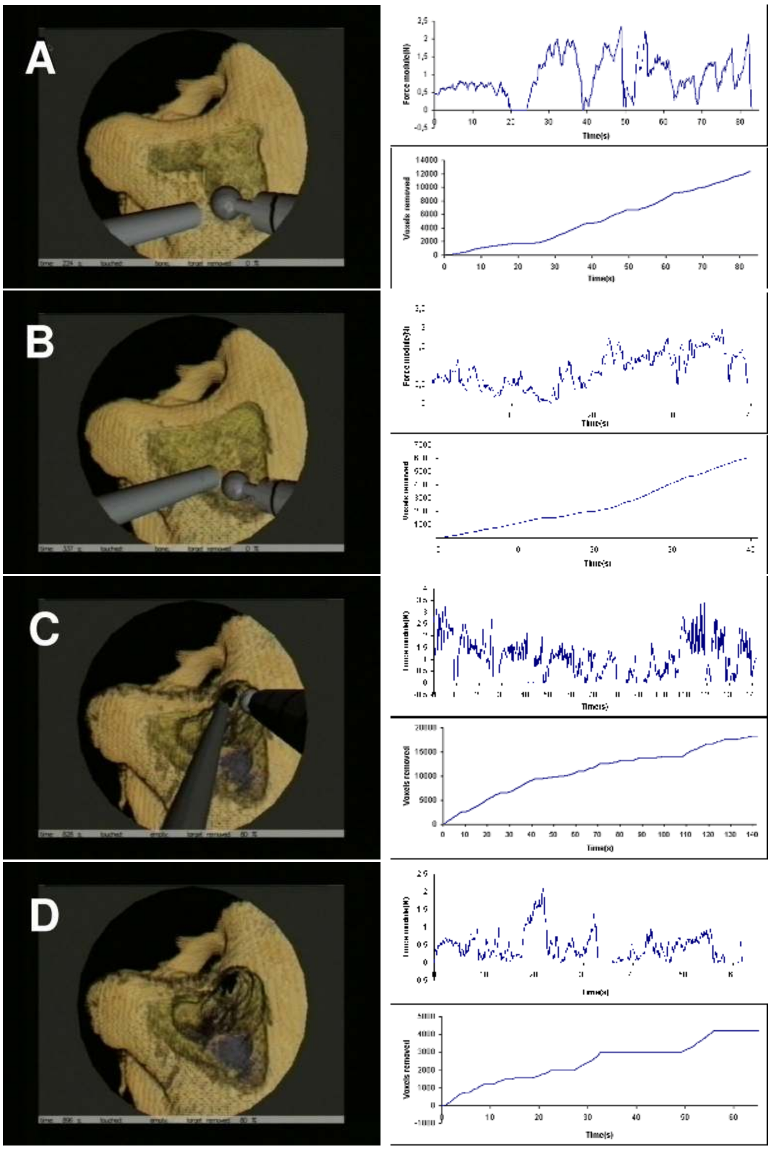


Figure 12.7.: Snapshot of the simulator (left) and plots of the force modulus and of the bone removal vs time (right) for the four mastoidectomy phases considered:A: cortex removal, B: cavity saucerization, C: identification of the mastoid antrum, D: localization of the facial recess

lies the basic landmark of the smoothly contoured, hard, labyrinthine bone of the horizontal semicircular canal. The localization of this canal allows exposure of the fossa incudis, the epitymphanum anteriorly and superiorly and the external genu of the facial nerve medially and inferiorly. Snapshot and plots relative to this step are shown in Figure12.7C. The final part of the basic mastoidectomy is represented in Figure12.7D. During this step several landmarks are identified, and also the facial recess area is discovered. Force and voxel removal plots show that each step in the surgical procedure can be characterized by different actions. In the first step, the force plot presents evident peaks and valleys due to the necessity of creating holes to start the bone removal. In the second step the force is more continuous and not too high. During the mastoid antrum exposure the force is irregular and reaches higher values, up to 3N. The removal rate is similar, about 10.000 voxel removed per second. Finally the last considered phase is characterized by large pauses where there is no voxel removal and even when removal is present its rate is lower than in the previous steps, indicating that critical sites have been reached and consequently burring movements are more careful and accurate. These facts can be pointed out just taking statistical values relative to the considered steps displayed in figures 12.8. It is possible, for example, to distinguish two phases with high average values of force and bone removal and two with lower values. The two phases with high bone removal can be distinguished by the average burr velocity: in the mastoid cortex removal, where the user try to start new paths for the bone removal, the velocity is limited, while in the mastoid atrium exposure, where the user removes small quantities of material burr's movements are much faster. The cavity saucerization and the facial nerve identification phases, characterized by lower force values can also be distinguished by correlating with the burr bit movements speed. In fact, in the first phase the burr moves quickly along already determined paths, while in the second it is moved slowly – and carefully – since there is an high risk of damaging the facial nerve.

12.5. Conclusion

This chapter was aimed to describe the current state of testing and validation of our simulation system for temporal bone surgical procedures. We presented preliminary results of the analysis of experimental data acquired during validating session of the virtual surgical training system for middle ear surgery. Tests are performed by expert surgeons and trainees and data are acquired in a controlled environment. These data can provide to the surgical community use-

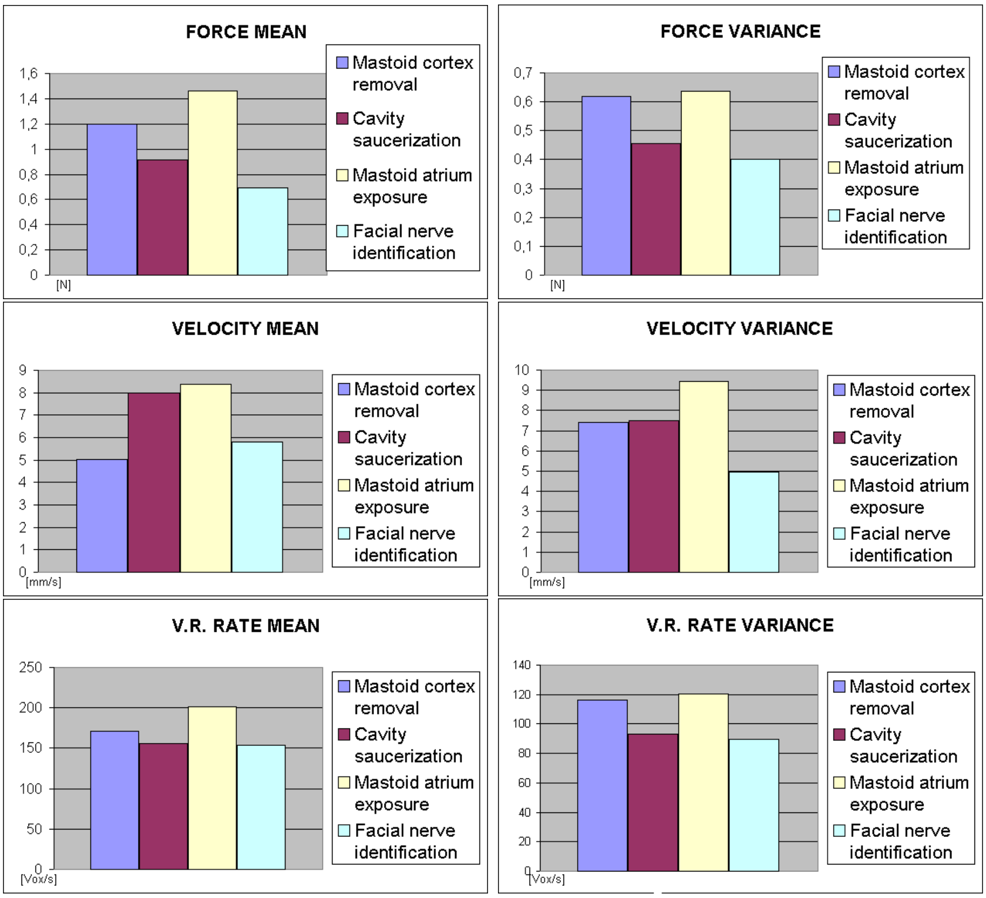


Figure 12.8.: Average value and variance of the force modulus, velocity and bone voxels removed during the four mastoidectomy phases considered

ful information to improve the training methods for critical surgical procedures involving bone dissection.

12.6. Bibliographical notes

The contents of this chapter are strictly related to reference [AGG$^+$03d], where we reported preliminary kinematic and dynamic analysis of simulated mastoidectomy sessions.

13. Tuning and Evaluation

In this chapter we provide the results on the tuning of the temporal bone surgical simulator using parameter values derived from experimental measurements, described in chapter 8 and on the comparison between these results and the previously used "domain expert" assigned values. Results indicate that the parameter values defined by the domain-experts are consistent with the experimentally derived values. Psychophysical testing indicates that the simulator is capable of rendering the basic material differences required for bone burring work and that some trained users preferentially associate a simulated temporal bone resin model with its real counterpart.

13.1. Introduction

The dynamic response of virtual reality surgical simulators is often controlled by physical models that are designed to capture the essential features of the anatomy, tissues and surgical tools involved. Given the real–time requirements imposed by this class of applications, and the current capabilities of hardware, these models are, usually, the result of a rather drastic simplification of the specific bio-mechanics involved, with their behavior controlled by a set of parameters that lump together details that would be otherwise uncomputable. The parameter set is then tuned to be consistent with the experience of surgeons proficient in the specific surgical procedure being simulated. Given the difficulties connected to direct *in-vivo* measurements, this is, usually, the only approach that can be followed; even though there has been significant recent progress in the development of instrumentation capable of direct, *in-vivo*, measurements of the bio-mechanical characteristics of soft tissues [BRK+03, GHC+02, KOG+03].

It remains, however, an interesting question if the two possible parameter definition approaches are actually compatible and, more specifically, given the limitations introduced by the computational algorithms and the devices used for haptic rendering, how much of the detail that can be obtained from direct physical measures will actually be usefully perceived by the user.

Here we report the results of our work on the tuning of a temporal bone surgical simulator using parameter values derived from experimental measurements and previously described in chapter 8, and on the comparison between these results and the values assigned by "domain expert" that we previously used. Specifically, we are interested in

- if the parameter values defined by domain-expert that we use to control the behavior of our bone-burring model are consistent with what would be obtained from direct experimental measures;
- how sensitive are humans to changes of parameters close to a reference value.

Our preliminary results indicate that parameter values defined by experts are consistent to experimental results, and that humans are not sensitive to change of parameters close to a reference value.

The rest of the chapter is organized as follows. After a brief review of related work, we introduce the principal concepts of psychophysics applied to haptic research, and the principal methods to derive threshold. Section 13.5 illustrates the psychophysical experiments we have performed thus far and their results. The chapter concludes with a discussion of the results obtained and a view of current and future work.

13.2. Related work

The development of advanced haptic interfaces requires information about the psychophysical capabilities of the human observer. Hesse and Wing [CA99] determined the displacement sensitivity function for human kinesthesis, movements with different peak velocities were imposed on the fingertip, using the PHANToM robot arm with a modified thimble attachment. Moy and others [MSTF00] quantified several perceptual capabilities of the human tactile system needed for teletaction, by performing psychophysics experiments to measure the amplitude resolution of the human tactile system, the effects of shear stress on grating orientation discrimination, and the effects of viscoelasticity (creep and relaxation) on tactile perception for static touch.

13.3. Haptics and Psychophysics

Psychophysics is a field of experimental psychology that uses specific behavioral methods to determine the relationship between the physical world and people's subjective experience of that world. Psychophysicists conduct scientific experiments that are carefully designed to let them figure out which physical parameter(s) actually determine a subjective perceptual dimension. Psychophysical methods allow to ask of the haptic system: how, and how well, do people sense and resolve intensive, spatial and temporal variations in mechanical (and thermal) inputs to the skin (the ”cutaneous” system) and to muscles, tendons and joints (the ”kinesthetic” system)? For example, psychophysical methods enable to determine the minimal intensity required to just detect the presence of, say, a static force, a vibration, a gap between two points, or a cool surface, applied to the skin, or the smallest movement of the hand that can just be detected. They allow also to determine the minimum change in stimulus intensity that is required to just notice a change in one's perception. But psychophysics don't use only physical values that produce threshold-level responses. It is also related to how people's sensations and perceptions grow in magnitude as a function of increasing physical intensity. The psychophysics target is to discover the nature of the mathematical function that best describes the relation between physical and perceptual parameters. Psychophysics provides a specific set of behavioral paradigms for addressing the types of physical parameters, namely sensitivity, resolving power, and the rate of growth of sensations. Behavioral techniques other than psychophysical ones can prove valuable as well in uncovering other important facts needed to know about human haptic processing. Psychophysical research can perform two general functions for haptic interfaces designers. First, the scientific information can be used to guide initial design considerations. For example, the psychophysical results can be used to select appropriate physical parameters and associated values for an interface system. Results from basic research on human haptics highlight both the strengths and the limitations of using the haptic system to operate a sensory interface. Such information allows to match the critical input/output parameters that underly human processing to the specific demands imposed by some haptic interface system. Some critical questions are: which type(s) of haptic information can or should be delivered? what are appropriate site(s) of delivery? and which are the best ways to display haptic information? While most designers of haptic interfaces have initially been forced to deal with relatively low-level sensory and motor issues, they should also be aware of additional cognitive influences (e.g., context, previous training, past knowledge, etc.) that may alter operator performance. The second function provided by psychophysics and, more generally, by the scientific study of biological touch is the set of formal experimental tools that are available for evaluating how well operators perform with the haptic interface. Some important issues to think about relate to making sure to develop an adequately broad and appropriate set of performance tests for assessing your system, and to gathering data in a scientifically appropriate way to ensure the validity and generality of conclusions.

13.4. Psychophysical Methods

Much (if not most) work in psychophysics has centered on determining how sensitive a sensory system is. This is determined by measuring how much of a particular stimulus is required to reliably detect that stimulus. In olden days one spoke of brides being carried across the threshold, or entrance, of a home. A sensory threshold is also a kind of entrance: it represents the entrance of a stimulus into sensory existence. The threshold for a particular light stimulus is that intensity which allows it to be “just seen”, while the threshold for a particular haptics stimulus is that intensity which allows it to be “just felt”. Thresholds are stochastic (that is, probabilistic, or variable), either because threshold actually changes over time (say because the general level of neural excitability changes), and/or because a variable amount of added ”equivalent stimulus” (called noise) is added to the detecting mechanisms on different series of trials. In any event, the modern concept of the threshold is that it is a probability that a (Y) response will be given.

13.4.1. Method of Limits

There are a variety of ways to measure threshold. A straightforward was is called the **Method of Limits**. Here, a stimulus is either gradually increased (Ascending Series) or decreased (Descending Series) in intensity, and the subject indicates on each trial (on each presentation) whether the stimulus was ”seen” or ”not seen” (or felt, or heard, or smelled, etc.). Series are often run in alternate ascending and descending directions. Graphs which plot a behavioral response (% detection) as a function of a physical variable (i.e., stimulus intensity) are referred to as psychometric functions. In tasks requiring a simple Yes or No response such as the one we are exploring, threshold is usually computed as the stimulus intensity yielding a 50% ”Yes, I felt it” response rate, although the decision is arbitrary, and other criteria may occasionally be used (e.g., 40% or 65%). Follow the long horizontal dashed line in the figure above which traces the 50% ”Yes” response rate across until it intersects the solid curved line. This smooth curve through the data points is the best fit to the response data of the table above (solid symbols) of a theoretical function called a cumulative normal distribution, which is the integral of a normal distribution. A general name for the family of such S-shaped functions is an ogive.

13.4.2. Method of Adjustment

Note that in Method of Limits the subject simply responds ”Yes” or ”No” to stimuli whose strength has been set by the experimenter. Another technique in which the subject takes a more active role is called the Method of Adjustment. Here, it is the subject who controls of the intensity of the stimulus. The subject adjusts the intensity until the stimulus is judged to be (in the case of a visual stimulus) ”just visible”. Method of adjustment is a straightforward and convenient technique, particularly for tracking thresholds which are undergoing rapid change over time.

13.4.3. Method of Constant Stimuli

One problem with either the Method of Limits or the Method of Adjustment is that the subject knows which intensity of the stimulus to expect from trial to trial (e.g., either a slightly stronger stimulus, in ascending series, or a slightly weaker one, in descending series) a problem which can influence the measured threshold. A method designed to overcome this expectation problem is the Method of Constant Stimuli. Here the order of presentation of the stimulus is randomized, so the subject cannot anticipate the intensity of the stimulus on any given trial. All the various stimulus intensities are presented, and a table similar to that for the Method of Limits (above) can be constructed. From such a table, the percent (Y) responses can be plotted as a function of stimulus strength, and a psychometric function can be described.

13.4.4. Method of Forced-Choice

While the Method of Constant Stimuli eliminates some of the problems of the previous methods, it still possesses a troubling flaw: the point at which ”No” responses become ”Yes” responses is determined not only by the stimulus threshold, but also by the whim of the subject. That is, there is no control over the response criterion applied to the decision of whether a stimulus was seen or not. This is a problem because some subjects may be cautious types, who refrain from saying ”Yes” until the stimulus is clearly visible to them. Other subjects are risk-takers, often saying ”Yes” even if they’re not entirely sure. Subjects may even change response criteria within the course of an experiment. A method developed to circumvent the problem of response criteria is the Method of Forced-Choice. Here subjects are presented with two or more alternatives, and must select one on each trial even if the stimulus was not clearly seen. The choice can thus be coded as a criterion-free ”correct” or ”incorrect”. Alternatives can be presented sequentially (temporal forced-choice), or can be presented simultaneously (spatial forced-choice). There must be at least two alternatives, but there can be up to four or five. (More than four or five usually becomes too confusing for subjects). Forced-choice methods usually reveal lower thresholds than other techniques. When subjects are forced to choose, they usually make better than chance guesses even when they *feel* like they’re just guessing. The forced choice psychometric function also differs from the Yes/No function in that it does not fall to 0% for weak stimuli. Instead, the worst performance is guessing, which in a 2AFC task is 50%. It is conventional to take the performance level half-way between guessing and perfect as the criterion for threshold in a 2AFC task (this is 75% correct in a 2AFC experiment). Forced-choice paradigms can be used to measure the sensory capacities of non-verbal subjects (human or animal), since subjects can be operantly conditioned to select the ”odd” stimulus from an array of three or more. The difference between the ”odd” stimulus and the others can be adjusted to smaller and smaller amounts while performance is tracked.

13.5. Psychophysical experiments

In order to evaluate the feel of the simulator in an objective manner we performed a series of psychophysical experiments. In this section we present their preliminary results.

13.5.1. Differentiating between virtual materials

Experts can feel the difference between real PVC and bone and the same subjective ability ought also apply to users of our simulator. In particular, two different tactile cues are known to be used in distinguishing real materials: (1) the tactile feedback received when probing a material’s surface; (2) the burring effect received when drilling through it. We investigated whether users could distinguish between simulated PVC and simulated bone using either of these two perceptual cues. We selected 20 volunteer subjects with no previous experience with the simulator. After an initial phase of familiarization with the simulator, each subject was exposed to two sequences of 12 trials. In the first sequence of trials the subject was asked to simply probe the surface with the drill, while in the second sequence they were asked to drill into the interior of the material. A 2AFC (**Two-Alternative-Forced-Choice**) design was followed whereby each trial is divided into two equal 20s intervals. In each interval the simulator rendered one of the two chosen material samples A or B. The subject's task was to indicate whether the material felt the same or different. The simulator was programmed to present samples in a random sequence while achieving an equal number of trials presenting the four sample pairs: AA, AB, BA, BB. The response of each subject to each test was recorded. The mean scores (out of 12) over all 20 subjects were as follows:

1. Probing surface: 9.3 ± 0.4

2. Drilling material: 9.6 ± 0.5

Both of these are significantly above chance response level (p=0.001). They correspond to about a 75% correct level with some individuals obtaining perfect results and others worse – see histograms below – as might be expected for a group of naive users. Clearly, our intention is that with further training most users would progress toward a near perfect score.

Fig. 13.1 histograms the results of the test. Note how our tester population divides itself between persons with different levels of ability.

As a comparison, we asked, using the same methodology described above, the best 12 subjects coming from the previous experiment to differentiate between the haptic simulation of Pettigrew temporal bone and real bone materials, see fig. 13.2. The data indicate that the volunteers were incapable of differentiating between the two materials.

13.5.2. Associating virtual to real materials

In order to investigate whether the simulator captures some of the physical attributes of the materials that are used by users to perceive a difference between them we conducted a final experiment. This attempted a direct comparison between a real bone milling and the simulated experience. Again a 2AFC design was employed to nullify potential response bias. Here the task was necessarily more complicated since the user had to "keep in mind" the feel of the real material while comparing it to one of two simulated samples. We found that naive users generally found this task too confusing to participate usefully. Thus we limited subjects to those that had participated in

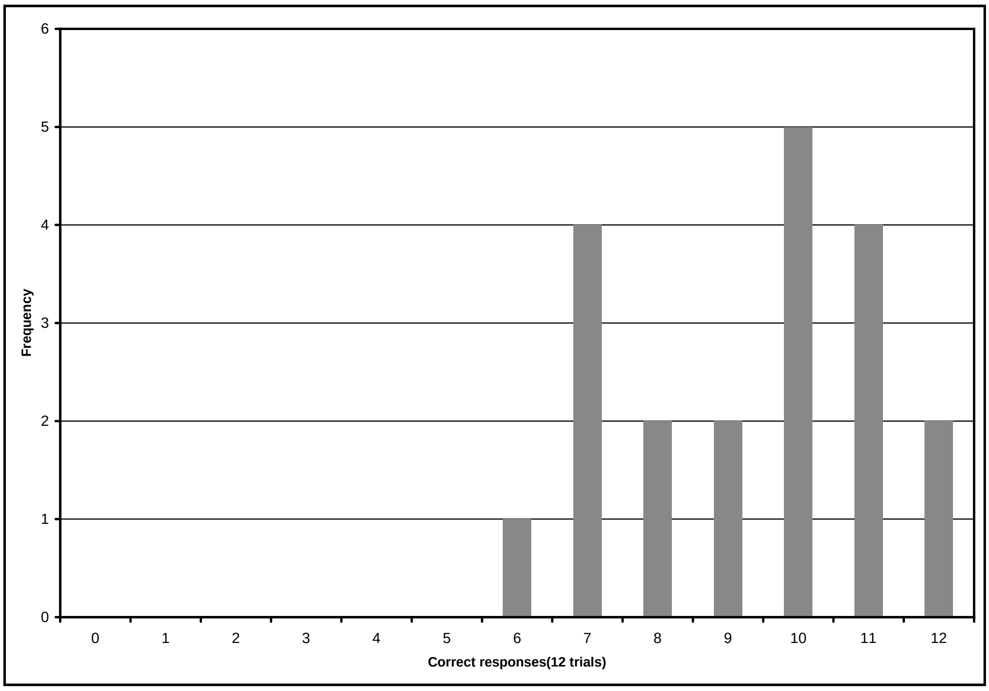


Figure 13.1.: **PVC vs Bone differentiation results**: the histogram represents results of the psychophysical tests performed by haptic simulation of PVC and bone materials. The x axis represents the score of correct responses out of 12 trials while the y axis represents the number of subjects that totalized that score. In each interval the simulator rendered one of the two chosen material samples A or B. The subject's task was to indicate whether the material felt the same or different.

the previous experiment and therefore had a good grounding in the simulated material difference. Each trial comprised three separate intervals, with the last two lasting 10 seconds, while the length of the first was at subject's discretion.

1. subjects burred the real sample of Pettigrew temporal bone
2. subjects used the simulator to burr sample A
3. subjects used the simulator to burr sample B.

The actual task was to say which of the simulated samples A or B felt most like the real sample. In each trial the simulated samples were PVC and Pettigrew temporal bone material but presented in random order. In all, five subjects each performed ten trials. The mean

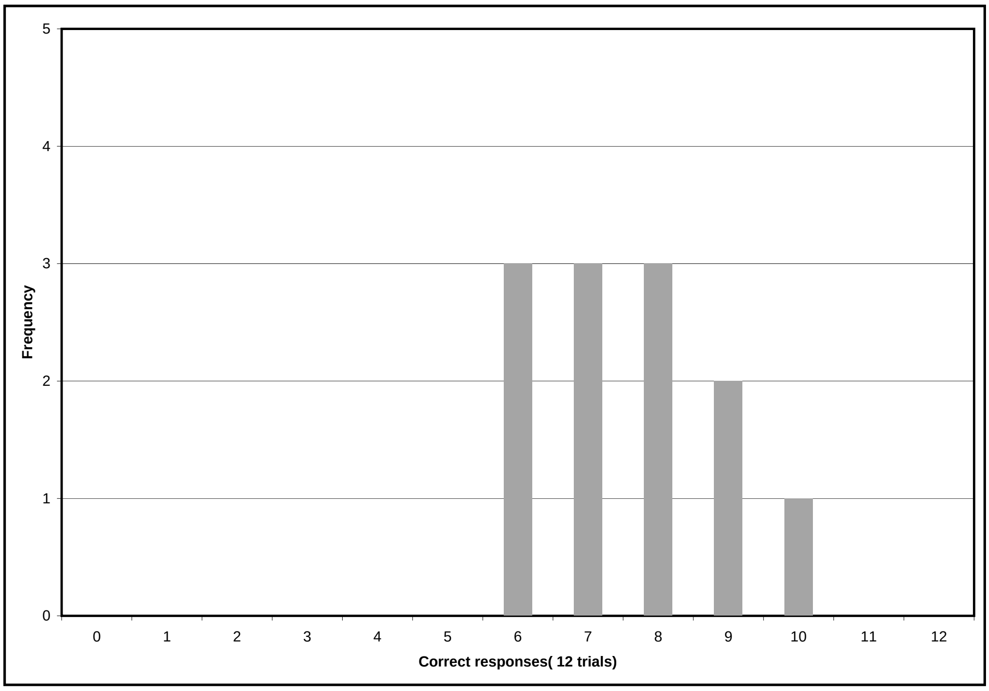


Figure 13.2.: **Pettigrew Temporal Bone vs Real Bone differentiation results**: the histogram represents results of the psychophysical tests performed by haptic simulation of Pettigrew temporal bone and real bone materials. The x axis represents the score of correct responses out of 12 trials while the y axis represents the number of subjects that totalized that score. In each interval the simulator rendered one of the two chosen material samples A or B. The subject's task was to indicate whether the material felt the same or different.

score (out of 10) over all 5 subjects was 9.4 ± 0.4. This is equivalent to 94% correct level and is clearly significantly above chance level (p=0.0001). This provides promising preliminary evidence that trained users are able to perceive the correspondence between real and simulated bone materials.

13.6. Discussion

In chapter 8 we described the experimental calibration of the physically-based haptic model of the surgical burr. To put experimental results in perspective, we have then evaluated, via psychophysical testing, the simulator rendition of three virtual materials defined, respectively, by the parameters values for the mastoid region of human temporal bone, the Pettigrew resin model and PVC. Our preliminary psychophysical results indicate that subjects can easily differentiate between the virtual temporal bone and PVC while even the best between them cannot distinguish between the virtual models of the human temporal bone and the Pettigrew plastic one.

Finally, we attempted a direct comparison between the milling of a real Plastic Pettigrew model, and the simulated rendition of it and PVC. The results provide promising preliminary evidence that trained users are able to perceive the correspondence between real and simulated Plastic temporal bone material.

We are currently in the process of designing a set of more detailed psychophysical tests in order to establish further the correspondence between the burring of real and simulated materials.

Concurrently we are working on defining metrics appropriate to the performance analysis of complete training sessions.

13.7. Bibliographical notes

This chapter reviews and expands the contents of reference [ABG$^+$04], where we provide the results on the tuning of the temporal bone surgical simulator using parameter values derived from experimental measurements and on psychophysical testing.

14. Conclusion and Discussion

This chapter reviews and discusses the principal achievements of the work.

14.1. Introduction

In this dissertation, we discussed a real–time haptic and visual implementation of a bone cutting burr, that it is being developed as a component of a training simulator for temporal bone surgery. The specific target of the simulator is mastoidectomy, a very common operative procedure that consists in the removal, by use of the burring tool, of the mastoid portion of the temporal bone. The importance of computerized tools to support surgical training for this kind of intervention has been recognized by a number of groups, which are currently developing virtual reality simulators for temporal bone surgery. Our work is characterized by the use of patient-specific volumetric object models directly derived from 3D CT and MRI images, and by a design that provides realistic visual and haptic feedback, including secondary effects, such as the obscuring of the operational site due to the accumulation of bone dust and other burring debris. The need to provide real–time feedback to users, while simulating burring and related secondary effects, imposes stringent performance constraints. Our solution is based on a volumetric representation of the scene, and it harnesses the locality of the physical system evolution to model the system as a collection of loosely coupled components running in parallel on a multi-processor PC platform. This chapter reviews and discusses the objectives and achievements of the work, and it is organized as follows: section 14.2 describes the principal objectives of our work, while section 14.3 lists the principal results. Finally section 14.4 provides informations about future work directions.

14.2. Objectives

The principal objective of the work was to show that is possible to render realistic haptic sensations of the contact between surgical burr and biological material, and that this kind of sensations are useful for the realization of virtual reality based training systems. The kernel of the thesis has been the derivation of simple physically based mathematical models that can replicate the behavior of a spherical rotating burr during typical surgical tasks. These models rely on a few parameters, and with the implementation and the employment of an experimental facility, we empirically derived the values of these parameters for materials of interest. The experimental results have been also compared with domain expert indications, and they showed that subjective tuning provided values close to those derived from experimental calibration. For this reason, in cases where experimental calibration is not practicable or too much complex, we think that the employment of subjective calibration techniques is to be considered a correct way for tuning haptic simulations. Finally, an intensive series of psychophysical testing showed that our physically-based haptic model of a surgical burr is capable of rendering different materials. Moreover, this dissertation described a virtual reality training system containing the haptic models, and provided details about the design and implementation, showing that the realistic immersing sensations are obtained by the employment of accurate visual volume rendering as well as realistic physically-based visual secondary effects. In order to obtain immersing sensations timing requirements needed to be satisfied, and this fact imposed constraints in the system design. The existing prototype is currently under evaluation, and we provided preliminary results, obtained during virtual training sessions of ENT surgeon trainees.

14.3. Achievements

The principal achievements obtained during the work are the following:

- physically-based haptic models of the behavior of a spherical surgical burr during bone dissection tasks;

- adaptive techniques for real-time haptic and visual simulation of bone dissection, that exploit a multi-resolution representation of the bone characteristic function to adaptively trade simulation quality with speed

- a custom experimental facility, used to characterize the behavior of materials of interest for surgical burring, including temporal bone specimens, and plastic replies commonly used for training purposes;
- the design and implementation of a surgical training system prototype containing the implementation of the haptic model of the cutting burr, an hardware-accelerated volume renderer, as well as a simulation of the visual secondary effects involved in the surgical specialty considered;
- the design and implementation of psychophysical experiments, to investigate the tactile cues used in distinguishing real materials.

14.4. Discussion and future work

A running prototype of the bone dissection simulator was realized, and it can be considered the principal tangible result of the work. The system is based on patient-specific volumetric object models derived from 3D CT and MR imaging data. Real-time feedback is provided to the trainees via direct volume rendering and haptic feedback. The performance constraints dictated by the human perceptual system are met by exploiting parallelism via a decoupled simulation approach on a multi-processor PC platform. The end-user evaluation of the simulator was based on three groups of users characterized as: experts (Senior ENT Surgeons); with theoretical knowledge (residents); no previous experience. Complete session traces were acquired during the training sessions and have been later analyzed and compared. The system demonstrated realistic haptic and visual rendering, and the ability of simulating complete mastoidectomy procedures. However, in order to improve the realism of the dust/fluid simulation, we plan to introduce in the next version of the simulator a more sophisticated treatment of particle collisions and thus to remove limitations such as the single particle per voxel constraint of the current implementation. One major limitation identified by end users is the resolution of the dataset, which is considered enough for early training but does not have enough anatomical detail to allow recognition of fine features [SWB$^+$02].

Moreover, the results of our preliminary psycho-physical testing provide promising preliminary evidence that trained users are able to perceive the correspondence between real and simulated Plastic temporal bone material. We are currently in the process of designing a set of more detailed psycho-physical tests in order to establish further the correspondence between the burring of real and simulated materials. Concurrently we are working on defining metrics appropriate to the performance analysis of complete training sessions. We plan to design performance psychophysical tests in order to find ways for measuring and assessing surgical skills.

14.5. Bibliographical notes

Most of the scientific results obtained during the work also appeared in related publications. Specifically, reference [JJT+ar] is a summary of the state-of-the-art in the subject; references [JTP+01, AGG+02b] provide a general overview of the project, mostly covering pre-operative planning; reference [AGG+02a] focuses on the human factor analysis; reference [AGG+02c] presents an implementation of visual and haptic simulation of bone dissection based on a ”first principles” model while reference [AGG+03b] describes adaptive techniques derived from the haptic and visual simulation. Finally reference [AGG+03a] focuses on the fitting of the haptic model to the experimental data, and reference [ABG+04] describes the tuning and evaluation of the haptic model.

Bibliography

[ABG+04] M. Agus, G. J. B. Brelstaff, A. Giachetti, E. Gobbetti, G. Zanetti, A. Zorcolo, B. Picasso, and S. Sellari Franceschini. Physics-based burr haptic simulation: tuning and evaluation. In *Proceedings of 12th Symposium on Haptic Interfaces For Virtual Environment And Teleoperator Systems*, Chicago - IL, March 2004. Accepted for publication.

[Ada99] R. J. Adams. *Stable Haptic Interaction with Virtual Environments*. PhD thesis, University of Washington, September 1999.

[AGG+02a] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, Nigel W. John, and Robert J. Stone. Mastoidectomy simulation with combined visual and haptic feedback. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, and D. Stredney, editors, *Medicine Meets Virtual Reality 2002*, pages 17–23. IOS Press, January 2002.

[AGG+02b] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. A multiprocessor decoupled system for the simulation of temporal bone surgery. *Computing and Visualization in Science*, 5(1), 2002.

[AGG+02c] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference*, pages 209–216. IEEE Computer Society Press, February 2002.

[AGG+03a] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Bruno Picasso, Stefano Sellari Franceschini, Gianluigi Zanetti, and Antonio Zorcolo. A haptic model of a bone-cutting burr. In editor J. D. Westwood, editor, *Medicine Meets Virtual Reality 2003*, pages 4–10, Amsterdam, The Netherlands, January 2003. IOS.

[AGG$^+$03b] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Adaptive techniques for real time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference*, pages 102 –109, Conference held in Los Angeles, CA, USA, March 22–26, March 2003.

[AGG$^+$03c] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. *Presence: Teleoperators and Virtual Environments*, 12(1):110–122, February 2003.

[AGG$^+$03d] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Tracking the movement of surgical tools in a virtual temporal bone dissection simulator. In *Surgery Simulation and Soft Tissue Modeling*, Lecture Notes in Computer Science, pages 102–109. Springer-Verlag Inc., New York, NY, USA, June 2003. To appear.

[AH99] R. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, 1999.

[AHJ$^+$01] Robert S. Allison, Laurence R. Harris, Michael Jenkin, Urszula Jasiobedzka, and James E. Zacher. Tolerance of temporal delay in virtual environments. In *Proceedings of the IEEE Virtual Reality 2001 International Conference*, volume 2, pages 247–254, March 2001.

[AJMSP01] Lecuyer A., Bruckhardt J-M., Coquillart S., and Coiffet PH. Boundary of illusion: an experiment of sensory integration with a pseudo-haptic system. In *Proceedings of VR2001*, pages 115–122, Yokohama, Japan, 2001.

[Ant] Patrick J. Antonelli. *Temporal Bone Dissection Manual*. University of Florida.

[AS96] Ricardo S. Avila and Lisa M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the conference on Visualization '96*, pages 197–ff. IEEE Computer Society Press, 1996.

[Bae98] Andreas Baerentzen. Octree-based volume sculpting. In *IEEE Visualization*, 1998.

[BGP$^+$02] Lionel Birglen, Clment M. Gosselin, Nicolas Pouliot, Bruno Monsarrat, and Thierry Lalibert. Shade, a new 3-dof haptic device. *IEEE Transactions on Robotics and Automation*, 18(2):166–175, 2002.

[BHB99] P. Berkelman, R. Hollis, and D. Baraff. Interaction with a realtime dynamic environment simulation using a magnetic levitation haptic interface device. In *IEEE International Conference on Robotics and Automation*, pages 3261 – 3266, May 1999.

[Bli94] James F. Blinn. Jim Blinn's corner: Compositing. 1. Theory. *IEEE Computer Graphics and Applications*, 14(5):83–87, September 1994.

[BN98] M. Bro-Nielsen. Finite element modeling in medical vr. *Journal of the IEEE*, 86(3):490–503, 1998.

[BRK$^+$03] J.D. Brown, J. Rosen, Y. S. Kim, L. Chang, M. N. Sinanan, and B. Hannaford. In-vivo and in-situ compressive properties of porcine abdominal soft tissues. In *Medicine Meets Virtual Reality 2003*, volume 94, pages 26–32. IOS Press, January 2003.

[BSWS01] Jason Bryan, Don Stredney, Greg Wiet, and Dennis Sessanna. Virtual temporal bone dissection: A case study. In *IEEE Visualization*, pages 497–500, 2001.

[BUB$^+$01] Iman Brouwer, Jeffrey Ustin, Loren Bentley, Alana Sherman, Neel Dhruv, and Frank Tendick. Measuring in vivo animal soft tissue properties for haptic modeling in surgical simulation. In J.D. Westwood et al., editor, *Medicine Meets Virtual Reality 2001*, pages 69–75. IOS Press, 2001.

[Bur00] Grigore C. Burdea. Haptic feedback for virtual reality. *special issue on Virtual Prototyping, International Journal of Design and Innovation Research*, 2:17–29, 2000.

[BV75] A. Bray and V. Vicentini. *Meccanica sperimentale*. Ed. Levrotto and Bella, 1975.

[CA99] Hesse C.W. and Wing A.M. Phantom psychophysics: Movement detection at the human fingertip. In *Proceedings of the First PHANToM Users Research Symposium*, 1999.

[CdVL95] Jim X. Chen and Niels da Vitoria Lobo. Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations. *Graph. Models Image Process.*, 57(2):107–116, 1995.

[CFT02] M. C. Cavusoglu, D. Feygin, and F. Tendick. A critical study of the mechanical and electrical properties of the phantom(tm) haptic interface and improvements for high performance control. *Presence*, 11(5), October 2002.

[CFW99] Jim X. Chen, Xiadong Fu, and J. Wegman. Real-time simulation of dust behavior generated by a fast traveling vehicle. *ACM Trans. Model. Comput. Simul.*, 9(2):81–104, 1999.

[CGO02] Vanessa B. Chial, Stephanie Greenish, and Allison M. Okamura. On the display of haptic recordings for cutting biological tissues. In IEEE, editor, *Proceedings of the 10th International Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 80–87, 2002.

[Chi00] Charlotte M. Chiong. *Temporal Bone Course Dissection Manual.* Department of Otolaryngology - University of Philippines, Manila, third edition, 2000.

[CO57] Shaw M. C. and C. J. Oxford. On the drilling of metals – the torque and thrust in drilling. *Transations of ASME*, 79:139–148, 1957.

[Col94] J. Colgate. Issues in the haptic display of tool use. In *Proceedings of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 140–144, 1994.

[Com96] General Reality Company. 5th glovetm. Technical report, 5DT, 1996.

[Cra86] J. Craig. *Introduction to Robotics:Mechanics and Control.* Addison–Wesley, Reading, Massachusetts, 1986.

[DC95] B. W. Davies and W. B. Campbell. Inguinal hernia repair: See one, do one, teach one? *Annals of the Royal College of Surgeons of England*, 77:299–301, 1995.

[DCH88b] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *Computer Graphicd*, 22(4):51–58, August 1988.

[DCH98] B. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In Rosalee Wolfe, editor, *Significant Seminal Papers of Computer Graphics: Pioneering Efforts that shaped the Field*, pages 363–372, N.Y., 1998. ACM Press.

[DGN95] Gomez D., Burdea G., and Langrana N. Integration of the rutgers master ii in a virtual reality simulation. In *Proceedings of the Virtual Reality Annual International Symposium*, pages 198–202, 1995.

[DGT$^+$03] Hugo DesRosiers, Daniel Gomez, Jerry Tian, Marc Tremblay, and Raymond Yu. *CyberGrasp User's Guide v1.2*. Immersion Corporation, 801 Fox Lane San Jose, CA USA, 2003.

[EKE01] K. Engel, M. Kraus, and T. Ertl. High quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *EuroGraphics/SIGGRAPH Workshop on Graphics Hardware*, 2001.

[EMF02] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744. ACM Press, 2002.

[ERSS99] Adelstein D. Bernard Ellis R. Stephen, Young J. Mark and Ehrlich M. Sheryl. Discrimination of changes of latency during voluntary hand movement of virtual objects. In *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society*, pages 1182–1186, 1999.

[ESJ97] R.E. Ellis, N. Sarkar, and M.A. Jenkins. Numerical methods for the force reflection of contact. *ASME Transactions on Dynamic Systems, Modeling, and Control*, 119(4):768–774, 1997.

[ESV00] J. El-Sana and A. Varshney. Continuously-adaptive haptic rendering. *Virtual Environments*, pages 135–144, 2000.

[EYAE99] S. Ellis, M. Young, B. D. Adelstein, and S. M. Ehrlich. Discrimination of changes in latency during head movement. In *Proceedings of the Eighth International Conference on Human-Computer Interaction*, volume 2, pages 1129–1136, 1999.

[FCG00] Eric Ferley, Marie Paule Cani, and Jean-Dominique Gascuel. Practical volume sculpting. *The Visual Computer*, 16(8):469–480, 2000.

[FF01] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30. ACM Press, 2001.

[FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graph. Models Image Process.*, 58(5):471–483, 1996.

[Fol87] J. D. Foley. Interfaces for advanced computing. *Scientific American*, 10, 1987.

[Fow01] Dennis Fowler. Computers and surgery – the future? *Contemporary Surgery*, 57(10):486, Oct. 2001.

[FPRJ00] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 249–254. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[Fuc02] K. H. Fuchs. Minimally invasive surgery. *Endoscopy*, 34(2):154–159, Feb 2002.

[GASF94] A. García-Alonso, N. Serrano, and J. Flaquer. Solving the collision detection problem. *IEEE Computer Graphics and Applications*, 14(3):36–42, 1994.

[GH91] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 267–274. ACM Press, 1991.

[GHC$^+$02] S. Greenish, V. Hayward, V. Chial, A. Okamura, and T. Steffen. Measurement, analysis and display of haptic signals during surgical cutting. *Presence:Teleoperators and Virtual Environments*, 6(11):626–651, December 2002.

[GJC$^+$87] Zimmermann T. G., Lanier J., Blouchard C., S. Bryson, and Y. Harvill. A hand gesture interface device. In *Proceedings CHI'87 Human Factors in Computing Systems*, pages 189–192, New York, 1987. ACM.

[GM00] S. F. Gibson and B. Mirtich. A survey of deformable modelling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Laboratories, Cambridge, MA, 2000.

[GME$^+$00] A. Gregory, A. Mascarenhas, S. Ehmann, M. C. Lin, and D. Manocha. 6-dof haptic display of polygonal models. In *Proc. of IEEE Visualization Conference*, 2000.

[GPZT98] Enrico Gobbetti, Piero Pili, Antonio Zorcolo, and Massimiliano Tuveri. Interactive virtual angioscopy. In *Proceedings IEEE Visualization*, pages 435–438, Conference held in Research Triangle Park, NC, USA, October 1998. IEEE Computer Society Press.

[Hab86] R. N. Haber. Flight simulation. *Scientific American*, 7:96–103, 1986.

[HIT88] T. Harada, S. Ishii, and N. Tayama. Three-dimensional reconstruction of the temporal bone from histological sections. *Arch Otolaryngol Head Neck Surg*, 114:1139–1142, 1988.

[HJ95] J. Hollerbach and S. Jacobsen. Haptic interfaces for teleoperation and virtual environments. In *Workshop on Simulation and Interaction in Virtual Environments*, 1995.

[HK97] Taosong He and Arie Kaufman. Collision detection for volumetric objects. In *Proceedings of the conference on Visualization '97*, pages 27–ff. ACM Press, 1997.

[Hol03] Nahvi HollerBack. Haptic interfaces, 2003. http://www.cs.utah.edu/classes/cs6360/Nahvi/haptic.html.

[Iwa90] Hiroo Iwata. Artificial reality with force-feedback: development of desktop virtual space with compact master manipulator. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 165–170. ACM Press, 1990.

[JIMK03] Timo Jokela, Netta Iivari, Juha Matero, and Minna Karukka. The standard of user-centered design and the standard definition of usability: analyzing iso 13407 against iso 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*, pages 53–60. ACM Press, 2003.

[JJJ01] Gary W. Johnson, Richard Jennings, and Richard Jennigns. *LabView Graphical Programming*. McGraw-Hill Professional, 2001.

[JJT$^+$ar] A. Jackson, N. W. John, N. A. Thacker, E. Gobbetti, G. Zanetti, R. J. Stone, A. D. Linney, G. H. Alusi, and A. Schwerdtner. Developing a virtual reality environment for petrous bone surgery: a “state-of-the-art” review. *Journal of Otology & Neurology*, To appear.

[JLG92] Furness R. J., Wu C. L., and Ulsoy A. G. Dynamic modeling of the thrust force and torque for drilling. In *Proceedings of the American Control Conference*, pages 384–390, 1992.

[JP92] G. Jacazio and B. Piombo. *Meccanica Applicata alle Macchine*. Levrotto and Bella, 1992.

[JTP$^+$01] N. W. John, N. Thacker, M. Pokric, A. Jackson, G. Zanetti, E. Gobbetti, A. Giachetti, R. J. Stone, J. Campos, A. Emmen, A. Schwerdtner, E. Neri, S. Sellari Franceschini, and F. Rubio. An integrated simulator for surgery of the petrous bone. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2001*, pages 218–224. IOS Press, January 2001.

[KCY93] Arie Kaufman, Daniel Cohen, and Roni Yagel. Volume graphics. *Computer*, 26(7):51–64, July 1993.

[Kil00] Mark J. Kilgard. A practical and robust bump-mapping technique for todays gpus. Technical report, NVIDIA Corporation, July 2000.

[Kil01] Mark J. Kilgard, editor. *NVIDIA OpenGL Extension Specifications*. NVIDIA Corporation, May 2001.

[KJM+97] RB Kuppersmith, R Johnston, D Moreau, RB Loftin, and H Jenkins. Building a virtual reality temporal bone dissection simulator. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 5*, pages 180–186. IOS Press, January 1997.

[KOG+03] D. Kalanovic, M. P. Ottensmeyer, J. Gross, G. Buess, and S.L. Dawson. Independent testing of soft tissue visco-elasticity using indentation and rotary shear deformations. In J.D. Westwood et al. (Eds.), editor, *Proc. Medicine Meets Virtual Reality 11 (MMVR 2003)*, pages 137–143, 2003.

[KOLM02] Y. Kim, M. Otaduy, M. Lin, and D. Manocha. 6-dof haptic display using localized contact computations. In *Proc. of Haptics Symposium*, 2002.

[KRL+03] Piet Kommers, Steffan Rdel, Jan-Maarten Luursema, Bob Geelkerken, and Eelco Kunst. Virtual reality for medical competences. Technical report, University of Twente, The Netherlands, 2003.

[L.02] Gamelli R. L. Is it time to change the way we educate surgeons? *Contemporary Surgery*, 58:168, 2002.

[LC87] W. Lorensen and H. Cline. Marching cubes: A high resolution 3-d surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.

[Lev88] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[LL86] L. Landau and E. Lifshitz. *Theory of elasticity*. Pergamon Press, 1986.

[LM93] X. Li and J.M. Moshell. Modeling soil: Realtime dynamic models for soil slippage and manipulation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 361–368, 1993.

[LW94] G. R. Luecke and J. Winkler. A magnetic interface for robot-applied virtual forces. *Dynamic Systems and Control*, DSC-Vol 55-1:271–276, 1994.

[Max95] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.

[May99] Deborah J. Mayhew. *The usability engineering lifecycle: a practitioner's handbook for user interface design.* Morgan Kaufmann Publishers Inc., 1999.

[MB98] Moreyra M. and Hannaford B. A practical measure of dynamic response of haptic devices. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[MBA02] L. Moody, C. Barber, and T. N. Arvanitis. Objective surgical performance evaluation based on haptic feedback. In *Medicine Meets Virtual Reality 02/10, Studies in Health Technology*, volume 85, pages 304–310, 2002.

[MBD99] DVM Melvin B. Dennis, Jr. Alternative training methods ii: Incorporating inanimate surgical models. *RESOURCE*, 28(5), May 1999.

[MJ86] Rolfe J. M. and Staples K. J. *Flight Simulation.* Cambridge University Press, 1986.

[MJ03] Nakamura M. and L. Jones. An actuator for the tactile vest - a torso-based haptic device. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, 2003.

[MPT99a] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 401–408. ACM Press/Addison-Wesley Publishing Co., 1999.

[MPT99b] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degrees-of-freedom haptic rendering using voxel sampling. In Alyn Rockwood, editor, *Siggraph 1999*, Annual Conference Series, pages 401–408. ACM Siggraph, Addison Wesley Longman, 1999.

[MSG95] Tomasz Mazuryk, Dieter Schmalstieg, and Michael Gervautz. Zoom rendering: Improving 3-D rendering performance with 2-D operations. Technical Report CG, Institute of Computer Graphics, Vienna University of Technology, 1995.

[MSTF00] G. Moy, U. Singh, E. Tan, and R. Fearing. Human psychophysics for teletaction system design. *Haptics-e, The Electronic Journal of Haptics Research*, 1(3), 2000.

[Nel91] R. A. Nelson. *Temporal Bone Surgical Dissection Manual.* House Ear Institute, Los Angeles, second edition, 1991.

[NNHJ98] A. Nahvi, D. Nelson, J. Hollerbach, and D. Johnson. Haptic manipulation of virtual mechanisms from mechanical cad designs. In *Proc. of 1998 Conference on Robotics and Automation*, pages 375–280, 1998.

[OIN+03] A. M. Okamura, R. J. Webster III, J. T. Nolin, K. W. Johnson, and H. Jafry. The haptic scissors: Cutting in virtual environments. In *Proceedings of the 2003 IEEE International Conference on Robotics And Automation Taipei, Taiwan, September 14-19, 2003*, pages 828–834, Taipei, Taiwan, September 2003.

[OL03] Miguel A. Otaduy and Ming C. Lin. Sensation preserving simplification for haptic rendering. *ACM Transactions on Graphics (TOG)*, 22(3):543–553, 2003.

[Oxf55] C. J. Oxford. On the drilling of metals – basic mechanics of the process. *Transation of ASME*, 1955.

[PB95] Buttolo P. and Hannaford B. Pen based force display for precision manipulation of virtual environments. In *Proceedings VRAIS-95*, pages 217–225, Raleigh, NC, March 1995.

[Pet] Alastair Pettigrew. Pettigrew plastic temporal bones. http://www.temporal-bone.com/.

[PK99] Kyoung Shin Park and Robert V. Kenyon. Effects of network characteristics on human performance in a collaborative virtual environment. In *Proceedings of IEEE VR '99*, pages 104–111, 1999.

[PPT+02a] Bernhard Pflesser, Andreas Petersik, Ulf Tiede, Karl Heinz Hohne, and Rudolf Leuwer. Haptic volume interaction with anatomic models at sub-voxel resolution. In *10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems,Proc. Haptics 2002*, pages 66–72, 2002.

[PR97] D. K. Pai and L. M Reissel. Haptic interaction with multiresolution image curves. *Computer and Graphics*, 21(405–411), 1997.

[PV95] F. Pinciroli and P Valenza. An inventory of computer resources for the medical application of virtual reality. *Computers in Biology and Medicine*, 25:115–125, 1995.

[RKK97] D. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proc. of ACM SIGGRAPH*, pages 345–352, 1997.

[RPP$^+$01] Matthias Renz, Carsten Preusche, Marco Pötke, Hans-Peter Kriegel, and Ger Hirzinger. Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. In *Proceedings of Eurohaptic Conference*, 2001.

[RS99] J.K. Hodgins R.W. Sumner, J.F. O'Brien. Animating sand, mud and snow. *Computer Graphics Forum*, 18:1, 1999.

[RSEB$^+$00] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage rasterization. In Stephan N. Spencer, editor, *Proceedings of the 2000 SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, pages 109–118. ACM Press, August 21–22 2000.

[Sab88] Paolo Sabella. A rendering algorithm for visualizing 3D scalar fields. *Computer Graphics*, 22(4):51–58, August 1988.

[Sat95] R. M. Satava. Medical applications of virtual reality. *Journal of Medical Systems*, 19:275–280, 1995.

[Sat96] Richard M. Satava. Advanced simulation technologies for surgical education. *Bulletin of the American College of Surgeons*, 81(7):77–80, 1996.

[Sch88] J. De Schutter. Improved force control laws for advanced tracking applications. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1497–1502, 1988.

[Sta99] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.

[Sto01] Bob Stone. A human-centred definition of surgical procedures. Technical Report IERAPSI (IST-1999-12175) Deliverable D2(part1), MUSE Virtual Presence, 2001.

[SWB$^+$02] D. Stredney, G. Wiet, J. Bryan, D. Sessanna, J. Murakami, O. Schamllbrock, K. Powell, and B. Welling. Temporal bone dissection simulation – an update. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, and D. Stredney, editors, *Medicine Meets Virtual Reality 2002*, pages 507–513. IOS Press, January 2002.

[SZP89] D. Sturmin, D. Zeltzer, and S. Pieper. Hands-on interaction with virtual environments. In *Graphics Symposium on User Interface Technology*, Williamsburg, 1989.

[TG70] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill, 1970.

[TK94] Massie T.H. and Salisbury K. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interface for Virtual Environments and Teleoperator Systems*, Chicago, IL, 1994.

[UK88] Craig Upson and Michael Keeler. V-BUFFER: Visible volume rendering. *Computer Graphics*, 22(4):59–64, August 1988.

[VK93] R. Volpe and P. Khosla. Computational considerations in the implementation of force control strategies. *Journal of Intelligent and Robotic Systems: Theory and Applications. Special Issue on Computational Aspects of Robot Kinematics, Dynamics, and Control.*, 1993.

[WBS$^+$00] G. Wiet, J. Bryan, D. Sessanna, D. Streadney, P. Schmalbrock, and B. Welling. Virtual temporal bone dissection simulation. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2000*, pages 378–384. IOS Press, January 2000.

[WE98] Rüdiger Westermann and Thomas Ertl. Efficiently using graphics hardware in volume rendering applications. In

Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 169–178. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[WK95] Sidney W. Wang and Arie E. Kaufman. Volume sculpting. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff. ACM Press, 1995.

[WMG98] Craig M. Wittenbrink, Thomas Malzbender, and Michael E. Goss. Opacity-weighted color interpolation for volume sampling. In *IEEE Symposium on Volume Visualization*, pages 135–142. IEEE, ACM SIGGRAPH, 1998.

[YJ00] Zhuang Y. and Canny J. Haptic interaction with global deformations. In *Proceedings of International Conference on Robotics and Automation*, 2000.

[ZAG03] Antonio Zorcolo, Marco Agus, and Enrico Gobbetti. Visualizzazione volumetrica diretta interattiva con effetti di illuminazione mediante Register Combiner OpenGL. Technical Report CRS4 TR/, CRS4, Center for Advanced Studies, Research, and Development in Sardinia, Cagliari, Italy, June 2003.

A. PHANToM Haptic Interface Mathematica Notebook

The Mathematica scripts, used to find direct kinematics and dynamics equations for the PHANToM Haptic Interface manipulator are provided in following sections

A.1. Direct Kinematics

A.1.1. Matrix transformations

Local matrix transformations, to obtain manipulator direct kinematics, where B is the base, E is the end-effector, O is the global reference frame and i=1..4 are the local reference frames.

```
TB1[q1_]:={{Cos[q1],-Sin[q1],0,0},
{Sin[q1],Cos[q1],0,0},{0,0,1,0},{0,0,0,1}};
T12[q2_]:={{Cos[q2],-Sin[q2],0,0},
{0,0,-1,0},{Sin[q2],Cos[q2],0,0},{0,0,0,1}};
T23[q3_]:={{Cos[q3],-Sin[q3],0,l1},
{Sin[q3],Cos[q3],0,0},{0,0,1,0},{0,0,0,1}};
T34:={{1,0,0,l2},
{0,1,0,0},{0,0,1,0},{0,0,0,1}};
T0B:={{0,1,0,0},{0,0,1,l2},{1,0,0,-l1},{0,0,0,1}};
T4E:={{0,-1,0,0},{0,0,1,0},{-1,0,0,0},{0,0,0,1}};
%Computation of the transformation matrix, from base to 4:
TB4=Simplify[TB1[q1].T12[q2].T23[q3].T34]
%Variable transformations,to obtain the same direct
% kinematics of Kavusoglu and al.
q1=th1[t];
q2=th2[t];
q3=th3[t]-th2[t]-Pi/2;
%Computation of the transformation matrix  from global
%reference frame to end-effector, exactly the same of Kavusoglu
T0E=Simplify[T0B.TB4.T4E]
```

A.2. Motion equations

A.2.1. Inertial moments

We proceed by computing the inertial moments of the bodies, referring to joints axis and positions (1), (2),(3). The method of computing the inertial moments is the following: J_{ij} is the inertial moment of the body i related to rotation axis j. To compute these moments, we use the inertial matrix of the hollow cylinders, and the Huygens theorem to move the moments computed relatively to the center of mass, to the joint positions. The principal inertial moments of the rigid body 1, considered as an hollow cylinder with length l_1, radius R, thickness r, and mass m_1 are:

```
Jzz1=1/2 m1 ( R^2 + (R -r)^2 );
Jxx1 = 1/2 Jzz1  + m1 l1^2 / 12;
Jyy1 = Jxx1;
%Inertial matrix of the rigid body 1:
J1    = DiagonalMatrix[ {Jxx1,Jyy1,Jzz1}];
%Axis of the joint 1 in the respect of the principal
%body reference:
axis1= {Cos[th2[t]], 0, Sin[th2[t]]};
%Inertial moment of the body 1 in respect of the
%joint 1, expressed
%by using the Huygens theorem:
J11 = Simplify[axis1.J1.axis1];
%Axis of the joint 2 in the respect of the
%principal body reference:
axis2 = {1,0,0};
%Inertial moment of the body 1 in respect
% of the joint 1, expressed
%by using the Huygens theorem:
J12 = Simplify[axis2.J1.axis2];
```

The principal inertial moments of the rigid body 2, considered as an hollow cylinder with length l_2, radius R, thickness r, and mass m_2 are:

```
Jxx2 = m2 / 4 ( R^2 + (R -r)^2 + l2^2 / 3 );
Jyy2 = Jxx2;
Jzz2 = m2 / 2 ( R^2 + (R - r)^2 );
%Inertial matrix of the rigid body 2:
J2 = DiagonalMatrix[{Jxx2, Jyy2, Jzz2 }];
%Rotation axis directions of the rigid body 2,
%in respect of the principal reference system:
axis1bis = {Sin[th3[t]],0, Cos[th3[t]]};
axis3 = axis2;
%Inertial moments of the rigid body 2, in respect
%of joints 1,2 and 3,
%computed by using the Huygens theorem:
J21 = Simplify[axis1bis.J2.axis1bis];
J22 = Simplify[axis2.J2.axis2];
J23 = Simplify[axis3.J2.axis3];
```

A.2.2. Kinetic energy of the baricenter

```
%Computation of the center of mass contribution
% to the kinetic energy of the body 1.
%Center of mass position of body 1, in the respect of B
%reference system
G1 = TB1[q1].T12[q2].{l1/2,0,0,0};
%Differentiate the center of mass position, in the respect of
%lagrangian parameters th1,th2,th3:
dG1 = { D[G1,th1[t]], D[G1,th2[t]], D[G1,th3[t]]};
dT = {th1’[t],th2’[t],th3’[t]};
%Compute the velocity of body 1, in the respect of lagrangian
%parameters:
v1= dT.dG1;
%Compute the center of mass of body 2, and the velocity:
G2 = TB1[q1].T12[q2].T23[q3].{l2/2,0,0,1};
dG2 = {D[G2,th1[t]],D[G2,th2[t]],D[G2,th3[t]]};
v2 = dT.dG2;
```

A.2.3. Kinetic energy of the system

```
%Kinetic energy of the mechanical system, considered as
%composed by two cylinder connected with rotational joints,
%rigid body 1 and rigid body 2
Ec=Ec1+Ec2;
%The kinetic energy of rigid body 1 is expressed using the Konig
%theorem, as sum of  the contribution of center of mass,
%and the rotation energy relative to joint 1 and joint 2.
Ec1 = 1/2 m1 Simplify[v1.v1] + 1/2 Simplify[J11] th1’[t]ˆ2\
 + 1/2 Simplify[J12] th2’[t]ˆ2;
%The same expression is used to compute the kinetic contribution
%of the body 2.
Ec2 = 1/2 m2 Simplify[v2.v2] + 1/2 Simplify[J21] th1’[t]ˆ2 \
+ 1/2 Simplify[J22] th2’[t]ˆ2 + 1/2 Simplify[J23] th3’[t]ˆ2;
%Potential energy of the system
V=V1 + V2;
V1 = m1 g G1[[3]];
V2 = m2 g G2[ [3] ];
%Dissipated energy
%Dissipated  energy, at joints 1,2,3, related to a damping
%parameter d:
Ed = d ( th1’[t]ˆ2 + th2’[t]ˆ2 + th3’[t]ˆ2);
```

A.2.4. Lagrangian formulation of motion

```
%We can now write the Lagrangian equations, by differentiating
%the energies involved in our mechanical systems:
Q1 = Simplify[D[D[Ec,th1’[t]],t]-D[Ec,th1[t]]+D[Ed,th1’[t]]+D[V,th1[t]]];
Q2 = Simplify[D[D[Ec,th2’[t]],t]-D[Ec,th2[t]]+D[Ed,th2’[t]]+D[V,th2[t]]];
Q3 = Simplify[D[D[Ec,th3’[t]],t]-D[Ec,th3[t]]+D[Ed,th3’[t]]+D[V,th3[t]]];
Simplify[Expand[Ec]]
```

```
Simplify[Expand[V]]
%Lagrangian force components
%Torque at joint 1 ( related to t1 Lagrangian parameter )
Simplify[Expand[Q1]]
%Torque at joint 2 ( related to t2 Lagrangian variable )
Simplify[Expand[Q2]]
%Torque at joint 3 ( related to t3 Lagrangian variable)
Simplify[Expand[Q3]]
%Coefficient of the motion equation
Tau1 = Collect[Expand[Q1],{th1’’[t],th1’[t]}];
M11 = Coefficient[Tau1, th1’’[t]];
Tau2 = Collect[Expand[Q2],{th2’’[t],th2’[t]}];
M22 = Simplify[Coefficient[Tau2, th2’’[t]]];
M23 = Coefficient[Tau2,th3’’[t]];
A21 = th1’[t] Coefficient[Tau2,th1’[t]ˆ2];
A23 = th3’[t] Coefficient[Tau2,th3’[t]ˆ2];
A22 = Coefficient[Tau2,th2’[t]];
Tau3 = Collect[Expand[Q3],{th3’’[t],th3’[t]}];
M33 = Coefficient[Tau3, th3’’[t]];
M32 = Coefficient[Tau3, th2’’[t]];
A31 = th1’[t] Coefficient[Tau3, th1’[t]ˆ2];
A32 = th2’[t] Coefficient[Tau3, th2’[t]ˆ2];
A33= Coefficient[Tau3,th3’[t]];
M = {{Collect[M11,{m1,m2}],0,0},\
{0,Collect[M22,{m1,m2}],Collect[M23,{m1,m2}]}\
,{0,Collect[M32,{m1,m2}],Collect[M33,{m1,m2}]}};
A13 =  - A31;
A12 =  - A21;
A11 = 1 / th1’[t] (Tau1 - M11 th1’’[t] - A12 th2’[t] - A13 th3’[t] );
A = {{A11,A12,A13},{A21,A22,A23},{A31,A32,A33}};
Tau = {Tau1  , Tau2   , Tau3};
Th’’ = { th1’’[t] , th2’’[t] , th3’’[t]};
Th’  = { th1’[t] ,  th2’[t] , th3’[t]};
P    = - Simplify[Expand[ M . Th’’ + A. Th’ - Tau]];
Simplify [ Expand[P + M . Th’’ + A . Th’ - Tau]]
{0, 0, 0}
%Numerical example
%We chose to model the manipulator with two indentical links, with r=0.
%The same values are used in a WorkingModel model
m=0.062832;
l=0.16;
R=0.01;
g=9.81;
d=0;
r=R;
m1=m2=m;
l1=l2=l;
%These are the matrix coefficients that we obtain
MatrixForm[Simplify[M]]
Simplify[Expand[A]]
Simplify[P]
```

B. PHANToM Matlab Scripts

B.1. PHANToM Kinematics

B.1.1. Direct Kinematics

```
function [x] = phantom_dk( th )
% function [x] = phantom_dk( th )
% Computes the direct kinematics of PHANToM haptic interface
% x is the end-effector cartesian position, and th are
% the lagrangian parameters
% x, th have to be  3-vectors

% link lengths
l1 = 0.16;
l2 = 0.16;
nelems = size(th);
for i = 1:nelems(1)
   x(i,1) = l1 * sin( th(i,1) ) * cos( th(i,2) )...
      + l2 * sin( th(i,1)) * sin( th(i,3) );
   x(i,2) = l2 - l2 *  cos( th(i,3) )...
      + l1 * sin( th(i,2));
   x(i,3) = -l1 + l1 * cos( th(i,1) ) * cos( th(i,2) )...
      + l2 * cos( th(i,1)) * sin( th(i,3) );
end
```

B.1.2. Inverse Kinematics

```
function [th] = phantom_ik( x )
% function [th] = phantom_ik( x )
% Computes the inverse kinematics of PHANToM haptic interface
% x is the end-effector cartesian position, and th are
% the lagrangian parameters
% x, th have to be  3-vectors

% link lengths
l1 = 0.16;
l2 = 0.16;

% float tolerance
tol = 1.0e-10;
```

```
R = sqrt( x(1)ˆ2 + ( x(3) + l1 )ˆ2 );
r = sqrt( Rˆ2    + ( x(2) - l2 )ˆ2 );

th(1) = atan2( x(1), x(3) + l1);

alpha = atan2( x(2) - l2, R);

th(2) = acos( ( l1ˆ2 + rˆ2 - l2ˆ2)/( 2 * l1 * r) ) + alpha;
th(3) = acos( ( l1ˆ2 + l2ˆ2 - rˆ2)/( 2 * l1 * l2) )...
 + th(2) - pi/2;
```

B.1.3. Kinematics errors script

```
% plot_phantom_kinematics_errors

n_points = 51;
cube_length = 0.2;
z    = 0.0;
x    = - cube_length/2: cube_length/(n_points-1): cube_length/2;
y    = - cube_length/2: cube_length/(n_points-1): cube_length/2;
err  = zeros(n_points,n_points);

% test the kinematics by partitioning the lagrangian
% parameter space
for i = 1:n_points,
   for j = 1:n_points,
         p        = [x(i) y(j) z]’;
         th       = phantom_ik( p );
         p_est    = phantom_dk( th );
         dp       = (p - p_est’)/norm(p);
         err(i,j) = abs( dp’ * dp);
   end;
end;

figure;
h = surf(x,y,err);
colormap hot;
shading interp;
set(h,’EdgeColor’,’k’);
set(gca,’ZScale’,’log’);
light(’Position’,[-2,2,20]);
lighting phong;
material([0.8,0.0,0.0,30]);
set(h,’FaceColor’,[0.0 0.9 0],...
   ’BackFaceLighting’,’lit’);

xlabel(’X Axis’);
ylabel(’Y Axis’);
zlabel(’Relative error’);
title(’Position errors for Z = 0’);
```

```
print -djpeg -r200 phantom_kinematics_test;
```

B.2. PHANToM Dynamics

B.2.1. Ode Function

```
function F = phantom_ode(t,y)
%PHANTom manipulator lagrangian dynamics formulation
%Example, with l1 = l2 = 16cm, R = r = 1cm, rho = 1000Kg/m^3
%as implemented in the WM model used to test Lagrangian
%formulation

% Torques applied to model
tau       = [0.01, 0.14, 0.008];

% Mechanical system status vector
% composed by lagrangian coordinates and their
% derivatives
theta     = [y(1) y(2) y(3)];
theta_dot = [y(4) y(5) y(6)];

% Parameters
a1  = 0.000402125;
a2  = 0.00107155;
a3  = 0.000267298;
a4  = 0.00080425;

% Inertia terms
M = [ 0.00134513+a2*cos(2*y(2))-a3*cos(2*y(3))...
      -a4*sin(y(2)-y(3))+a4*sin(y(2)+y(3))   0   0;
     0   0.00228185   -a4*sin(y(2)-y(3));
     0   -a4*sin(y(2)-y(3)) 0.000537737];

% Coriolis terms
a11 = -2*a1*sin(y(2))*sin(y(3))*y(5)...
   -a2*sin(2*y(2))*y(5)+2*a1*cos(y(2))*cos(y(3))*y(6)...
      +a3*sin(2*y(3))*y(6);
   a12 =  -a1*cos(y(2)-y(3))*y(4)+...
      a1*cos(y(2)+y(3))*y(4)-a2*sin(2*y(2))*y(4);
   a13 =  a1*cos(y(2)-y(3))*y(4)...
      +a1*cos(y(2)+y(3))*y(4)+a3*sin(2*y(3))*y(4);
a21 =  -a12;
a31 =  -a13;
a23 =   a4*cos(y(2) - y(3))*y(6);
a32 =  -a4*cos(y(2) - y(3))*y(5);
A = [  a11  a12 a13; -a21 0 a23; -a13 a32 0];

% Gravity terms
```

```
P = [0, 0.147932*cos(y(2)), 0.0493106*sin(y(3))];

% ODE Function
csi   = inv(M)*(tau'-P'-A*theta_dot');
F = [ theta_dot, csi']';
```

B.2.2. Inverse Dynamics

```
% Phantom ode solver script
% to solve dynamic equations

% Input parameters
t_max     = 0.5;
filename  = 'G:\\program files\\Working Model 3D...
\\phantom_test.txt';

% Time scan
T_scans   = 0:0.001:t_max;

% Initial condition
Y0 = [ [ 0 0 0], [0 0 0]];

% Solve PHANToM Lagrangian Dynamics
% The problem definition is in phantom_ode.m
[T,Y]=ode15s('phantom_ode', T_scans, Y0);
Theta     = Y(:,1:3);
Theta_dot = Y(:,4:6);

% Plot joint orientation
figure(1);
h1 = plot(T, Theta);
title('PHANToM Joints rotations');
legend(h1,'\theta_1','\theta_2','\theta_3',3);

% Plot joint angular velocities
figure(2);
h2 = plot(T, Theta_dot);
title('PHANToM Joints angular speeds');
legend(h2,'\partial \theta_1','\partial \theta_2',
...'\partial \theta_3',3);

% PHANToM direct kinematics
X = phantom_dk( Theta );

%Plot end-effector cartesian positions
figure(3);
h3 = plot(T, X);
title('PHANToM end effector positions');
legend(h3,'X_e','Y_e','Z_e',3);
print -djpeg -r200 th_end_position;
```

```
%Load data from Working Model 3D
Data_wm  = load(filename);
T_wm     = Data_wm(:,1);
X_wm     = [Data_wm(:,3), Data_wm(:,4), Data_wm(:,2)];
figure(4);
h4 = plot(T_wm, X_wm,'LineWidth',2);
title('WorkingModel PHANToM positions');
legend(h4,'X_e','Y_e','Z_e',3);
xlabel('Time (s)');
ylabel('Position (m)');
print -djpeg -r200 wm_end_position;

%Compute errors
figure(5);
E = sqrt( (X-X_wm).*(X-X_wm));
h5 = semilogy(T_wm, E,'LineWidth',2);
title('Position relative errors');
legend(h5,'\epsilon_x','\epsilon_y','\epsilon_z',3);
xlabel('Time (s)');
ylabel('Error');
print -djpeg -r200 pos_error;
```

B.2.3. Direct Dynamics

```
% Script to test the direct dynamics of the PHANToM manipulator
n_points          = 2000;
t_initial         = 0.0;
t_final           = 2.0;
duration          = t_final - t_initial;
desc_displacement = 0.1;
theta_initial     = [ 0 0 0];
delta             = 0.25;
P_final           = [ 0.2 -desc_displacement -0.05];
theta_final       = phantom_ik(P_final);

theta_half        = ( theta_initial + theta_final )/2;
theta_dot         = 2 * theta_half/( duration *(1 - delta ));
t_left            = t_initial + delta*duration;
t_half            = t_initial + duration/2;
t_right           = t_final   - delta*duration;

theta_2dot        = ( theta_half + theta_dot*(t_left - t_half))...
   /(t_left-t_initial)^2;
theta_left        = theta_initial +...
 (t_left  - t_initial)^2*theta_2dot;
theta_right       = theta_final...
   - (t_final - t_right)^2*theta_2dot;

% Parameters
a1  = 0.000402125;
a2  = 0.00107155;
```

```
a3  = 0.000267298;
a4  = 0.00080425;

t                 =  zeros( n_points, 1);
theta             =  zeros( n_points, 3);
tau               =  zeros( n_points, 3);

for i = 1:n_points,
   t(i) =  t_initial +  (i - 1)/(n_points - 1)*duration;
   if t(i) <= t_left,
      theta(i,:)   = theta_initial+theta_2dot*(t(i)-t_initial)ˆ2;
      theta_2dot_i = theta_2dot;
      theta_dot_i  = 2 * theta_2dot * (t(i)-t_initial);
   elseif t(i)>t_left & t(i)<=t_right,
      theta(i,:) = theta_left+theta_dot*(t(i)-t_left);
      theta_2dot_i = [0 0 0];
   theta_dot_i  = theta_dot;
   else
      theta(i,:) = theta_final-theta_2dot*(t_final-t(i))ˆ2;
      theta_2dot_i = -theta_2dot;
   theta_dot_i  = 2 * theta_dot *(t_final-t(i)) ;
   end
   % Inertia terms
   M = [ 0.00134513+a2*cos(2*theta(i,2))...
         -a3*cos(2*theta(i,3))...
         -a4*sin(theta(i,2)-theta(i,3))...
         +a4*sin(theta(i,2)+theta(i,3))   0   0;
     0   0.00228185   -a4*sin(theta(i,2)-theta(i,3));
     0   -a4*sin(theta(i,2)-theta(i,3)) 0.000537737];

   % Coriolis terms
   a11 = -2*a1*sin(theta(i,2))*sin(theta(i,3))*...
      theta_dot_i(2)-a2*sin(2*theta(i,2))*theta_dot_i(2)...
      +2*a1*cos(theta(i,2))*cos(theta(i,3))*theta_dot_i(3)...
      +a3*sin(2*theta(i,3))*theta_dot_i(3);
   a12 =  -a1*cos(theta(i,2)-theta(i,3))*theta_dot_i(1)...
      +a1*cos(theta(i,2)+theta(i,3))*theta_dot_i(1)...
      -a2*sin(2*theta(i,2))*theta_dot_i(1);
   a13 =  a1*cos(theta(i,2)-theta(i,3))*theta_dot_i(1)...
      +a1*cos(theta(i,2)+theta(i,3))*theta_dot_i(1)...
      +a3*sin(2*theta(i,3))*theta_dot_i(1);
   a21 =  -a12;
   a31 =  -a13;
   a23 =   a4*cos(theta(i,2) - theta(i,3))*theta_dot_i(3);
   a32 =  -a4*cos(theta(i,2) - theta(i,3))*theta_dot_i(2);
   A = [  a11  a12 a13; -a21 0 a23; -a13 a32 0];

   % Gravity terms
   P = [0, 0.147932*cos(theta(i,2)), 0.0493106*sin(theta(i,3))];
   tau(i,:) = (M*theta_2dot_i’+A*theta_dot_i’+P’)’;

end
```

```
figure(1);
h1 = plot(t, theta,’LineWidth’,2);
legend(h1,’\theta_1’,’\theta_2’,’\theta_3’);
title(’PHANToM Joint Trajectory’);

figure(2);
h2 = plot(t, tau,’LineWidth’,2);
legend(h2,’\tau_1’,’\tau_2’,’\tau_3’);
title(’PHANToM torques’);
```

Curriculum Vitae

Marco Agus was born in Cagliari on December, 21th 1974. He studied Electronics Engineering at the University of Cagliari, where he received the Laurea (M. Sc. degree) in October 1999. Since 2000, he is a researcher at CRS4, in the Visual Computing Group. His current interests are related to computer graphics and scientific visualization, haptic simulations, and surgical simulation.

Publications

The following list contains published and accepted papers in international journals and conference proceedings, as well as technical reports and multimedia .

Journal Articles

[1] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. A multiprocessor decoupled system for the simulation of temporal bone surgery. *Computing and Visualization in Science*, 5(1):35–43, 2002.

[2] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. *Presence: Teleoperators and Virtual Environments*, 12(1):110–122, February 2003.

[3] Enrico Gobbetti, Leonardo Spanò, and Marco Agus. Hierarchical higher order face cluster radiosity for global illumination walkthroughs of complex non-diffuse environments. *Computer Graphics Forum*, 22(3):563–572, September 2003.

Book Contributions

[1] Marco Agus, Fabio Bettio, Enrico Gobbetti, and Luciano Fadiga. An integrated environment for steroscopic acquisition, off-line

3D elaboration, and visual presentation of biological actions. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, D. Stredney, and R. A. Robb, editors, *Medicine Meets Virtual Reality 2001 – Inner Space, Outer Space, Virtual Space*, pages 23–29. IOS, Amsterdam, The Netherlands, January 2001.

[2] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Bruno Picasso, Stefano Sellari Franceschini, Gianluigi Zanetti, and Antonio Zorcolo. A haptic model of a bone-cutting burr. In J. D. Westwood, editor, *Medicine Meets Virtual Reality 2003*, pages 4–10. IOS, Amsterdam, The Netherlands, January 2003.

[3] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, Nigel W. John, and Robert J. Stone. Mastoidectomy simulation with combined visual and haptic feedback. In J. D. Westwood, H. M. Hoffmann, G. T. Mogel, and D. Stredney, editors, *Medicine Meets Virtual Reality 2002*, pages 17–23. IOS, Amsterdam, The Netherlands, January 2002.

[4] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Tracking the movement of surgical tools in a virtual temporal bone dissection simulator. In *Surgery Simulation and Soft Tissue Modeling*, Lecture Notes in Computer Science, pages 102–109. Springer-Verlag Inc., New York, NY, USA, June 2003. To appear.

Conference Papers

[1] Marco Agus, Gavin Brelstaff, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, Antonio Zorcolo, Bruno Picasso, and Stefano Sellari Franceschini. Physics-based burr haptic simulation: tuning and evaluation. In *Proceedings IEEE Haptics*, Conference held in Chicago, IL, USA, April 2004. IEEE Computer Society Press.

[2] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Haptic simulation of bone dissection. In *Abstract – SIMAI 2002 Symposium on Methods and Applications in Advanced Computational Mechanics*, Conference held in Chia, CA, Italy, May 27–31, 2002, May 2002.

[3] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Real-time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference*, pages 209–216, Conference held in Orlando, FL, USA, March 24–28, February 2002. IEEE Computer Society Press.

[4] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Un sistema multiprocessore per la simulazione della chirurgia sull'osso temporale. In *Proceedings of the Second MIMOS Conference*, Conference held in Turin, Italy, October 28–29, 2002, October 2002.

[5] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. Adaptive techniques for real–time haptic and visual simulation of bone dissection. In *IEEE Virtual Reality Conference*, pages 102–109, Conference held in Los Angeles, CA, USA, March 22–26, March 2003. IEEE Computer Society Press.

[6] Andrea Giachetti, Marco Agus, Enrico Gobbetti, and Gianluigi Zanetti. Blood, dust and mud for a temporal bone surgical simulator. In *Proceedings of the 5th Conference of the European Society of Mathematical and Theoretical Biology, Poster sessions*, Conference held in Milan, Italy, July 2–6, 2002, July 2002.

Technical Reports

[1] Marco Agus, Fabio Bettio, and Enrico Gobbetti. Creating and presenting real and artificial visual stimuli for the neurophysiological investigation of the observation/execution matching system. Technical Report 00/, CRS4, Center for Advanced Studies, Research, and Development in Sardinia, Cagliari, Italy, June 2000.

[2] Marco Agus, Andrea Giachetti, Enrico Gobbetti, and Gianluigi Zanetti. IERAPSI surgical simulator software kernel. Deliverable D4.1, EU Project IERAPSI (IST-1999-12175), February 2002.

[3] Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. IERAPSI petrous bone surgical simulation platform. Deliverable D4.2, EU Project IERAPSI (IST-1999-12175), January 2003.

[4] Enrico Gobbetti, Riccardo Scateni, and Marco Agus. Exploring virtual prototypes using time-critical rendering. Technical Report CRS4 TR/, CRS4, Center for Advanced Studies, Research, and Development in Sardinia, Cagliari, Italy, November 1999.

[5] Enrico Gobbetti, Leonardo Spanò, and Marco Agus. Hierarchical higher order face cluster radiosity. Technical Report CRS4

TR/, CRS4, Center for Advanced Studies, Research, and Development in Sardinia, Cagliari, Italy, March 2002.

[6] Nigel W. John, Neil A. Thacker, Maja Pokric, Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, Robert J. Stone, Manfred Kummer, and Frederic Rubio. IERAPSI surgical procedures and implementation specification. Deliverable D2, EU Project IERAPSI (IST-1999-12175), July 2000.

[7] Antonio Zorcolo, Marco Agus, and Enrico Gobbetti. Visualizzazione volumetrica diretta interattiva con effetti di illuminazione mediante Register Combiner OpenGL. Technical Report CRS4 TR/, CRS4, Center for Advanced Studies, Research, and Development in Sardinia, Cagliari, Italy, June 2003.

Videos and Multimedia Productions

[1] Marco Agus. Studio e realizzazione di un browser vrml multirisoluzione. Video 82, CRS4, October 1999.

[2] Marco Agus, Fabio Bettio, Andrea Giachetti, Enrico Gobbetti, A.Zorcolo Gianluigi Zanetti, M.D'Anteo, S.Sellari Franceschini, , and S.Valori. Real and virtual surgical procedures on the temporal bone. Video 104, CRS4, October 2002. Presented at: IEEE Virtual Reality Conference, Conference held in Los Angeles, CA, USA, March 22-26, 2003.

[3] Enrico Gobbetti, Leonardo Span, and Marco Agus. Hierarchical higher order face cluster radiosity for global illumination walkthroughs of complex non-diffuse environments. Video 107, CRS4, February 2003. Presented at the Eurographics 2003 Conference.

[4] M.Agus, F.Bettio, A.Giachetti, E.Gobbetti, G.Zanetti, , and A.Zorcolo. Real-time haptic and visual simulation of bone dissection. Video 97, CRS4, September 2002. Presented at: IEEE Virtual Reality Conference, Conference held in Orlando, FL, USA, March 24-28, 2002.

[5] M.Agus, F.Bettio, A.Giachetti, E.Gobbetti, G.Zanetti, A.Zorcolo, and S.Sellari Franceschini. Extradural bone removal simulation. Video 103, CRS4, October 2002.

[6] Antonio Zorcolo, Enrico Gobbetti, and Marco Agus. Head tracked direct volume rendering with shading for medical data exploration. Video 112, CRS4, June 2003.

[7] Antonio Zorcolo, Enrico Gobbetti, and Marco Agus. Interactive direct volume rendering with shading for medical data exploration. Video 109, CRS4, June 2003.