DEVELOPMENT OF DENTAL EDUCATIONAL SIMULATION WITH HAPTIC DEVICE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

UMUT KOÇAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MEDICAL INFORMATICS

SEPTEMBER 2007

Approval of the Graduate School of Informatics

_____

Prof. Dr. Nazife Baykal

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Assoc..Prof. Dr. Erkan Mumcuoğlu

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science of Medical Informatics.

_____	_____
Assoc.Prof. Dr. Erkan Mumcuoğlu	Assist.Prof. Dr. E.İlhan Konukseven
Co-Supervisor	Supervisor

Examining Committee Members

Assoc.Prof.Dr. Yasemin Yardımcı	(METU II) _____
Assist.Prof.Dr. E.İlhan Konukseven	(METU ME) _____
Assoc.Prof.Dr.Erkan Mumcuoğlu	(METU II) _____
Assoc.Prof.Dr.Veysi İşler	(METU CENG) _____
Assist.Prof.Dr.Yiğit Yazıcıoğlu	(METU ME) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name: Umut Koçak

Signature : _____

ABSTRACT

DEVELOPMENT OF DENTAL EDUCATIONAL SIMULATION WITH HAPTIC DEVICE

Koçak, Umut

M.S., Department of Medical Informatics

Supervisor: Assist.Prof. Dr. E.İlhan Konukseven

Co-Supervisor: Assoc.Prof. Dr. Erkan Mumcuoğlu

September 2007, 89 pages

Virtual Reality (VR) applications in medicine had significant improvements. 3D visualization of various anatomical parts using advanced medical scanner images, anatomy education, surgery operation simulation, virtual simulator for laparoscopic skills, virtual endoscopy, psychotherapy and rehabilitation techniques are some of the VR applications in medicine. Integration of the haptic devices into VR applications increased quality of the systems. By using haptic devices, the user can not only feed information to the computer but can receive information from the computer in the form of a felt sensation on some part of the body.

In this thesis a dental education simulator is developed by using a computer, a monitor, a haptic device and stereoscopic devices. The entire jaw model, all teeth and decay is modeled in the virtual environment. It is possible to diagnose the decay and remove the decay region by using different dental instruments developed in the system. Different graphical rendering methods like Marching Cubes, Ray-casting on GPU are implemented and compared.

The system is used by dentists from METU Health Center and Ankara University and performance tests are applied to the system. By this system it is expected to develop a more realistic and effective preclinical education. Several advantages offered by the simulator include: an effective learning environment without undue fear of mistakes, facilitation of repetition, provision of opportunities to quantitatively assess student skills, rapid training environment without an instructor and lower the cost of dentist training.

Keywords: Virtual Reality, haptic, education simulator, dental education.

ÖZ

DİŞ HEKİMLİĞİ EĞİTİM SİMÜLATÖRÜ GELİŞTİRİLMESİ

Koçak, Umut
Yüksek Lisans, Tıp Bilişimi
Tez Yöneticisi: Yard.Doç. Dr. E.İlhan Konukseven
Ortak Tez Yöneticisi: Doç. Dr. Erkan Mumcuoğlu

Eylül 2007, 89 sayfa

Medikal alandaki sanal gerçeklik uygulamaları son yıllarda büyük gelişme göstermiştir. Gelişmiş görüntüleme tekniklerinden yola çıkarak elde edilen 3 boyutlu görselleştirmeler, anatomi eğitimi uygulamaları, cerrahi operasyon simülasyonları, sanal laparoskopi simülatörleri, sanal endoskopi, psikoterapi ve rehabilitasyon uygulamaları bunlardan bazılarıdır. Bu eğitim uygulamalarının daha gerçekçi olabilmesi için dokunma hissi veren "Haptik" cihazlar kullanılmaya başlanmıştır. "Haptik" cihazlar sıradan veri giriş araçlarının aksine kullanıcıya dokunma hissi aracılığıyla geri besleme de sağlarlar.

Bu tez kapsamında bir bilgisayar, bir monitör, “haptic” cihazı, özel gözlük enstrümanı,“stereovision”, kullanılarak diş hekimliği eğitimine yönelik bir simülasyon yazılımı geliştirilmiştir. Sanal ortamda tüm çene modeli, sağlam ve çürük dişler taklit edilmiştir. Sistem kullanılarak çürük tanısı yapılabilmekte ve ardından çürük bölge temizlenebilmektedir. “Yürüyen Küpler”, GPU kullanılarak Işın Fırlatma (Ray-Casting) gibi farklı görüntüleme yöntemleri uygulanmış ve mukayese edilmiştir.

Sistem geliştirildikten sonra O.D.T.Ü Sağlık ve Rehberlik Merkezi ve Ankara Üniversitesi’nden diş hekimleri tarafından denenmiş ve performans testleri yapılmıştır. Geliştirilen sistem sayesinde diş hekimliği preklinik eğitiminin çok daha gerçekçi ve etkili olmasını, öğrencilerin kliniğe daha hazır olarak çıkmalarını sağlamak mümkündür. Sistemin sağlayacağı temel avantajlar: hata yapma korkusu olmayan bir eğitim ortamı, zorlanılan uygulamaların tekrarlanabilmesi, öğrenci performanslarının nicel olarak değerlendirilebilmesi, gözetmen gerektirmeyen daha hızlı bir öğrenme ortamı ve eğitim masraflarının azaltılması olarak sayılabilir

Anahtar Kelimeler: Sanal Gerçeklik, “Haptic” Cihazı, Eğitim Simülatörü, Diş Hekimliği Eğitimi..

To My Family

ACKNOWLEDGMENTS

I express sincere appreciation to Assist.Prof. Dr. İlhan Konukseven for his guidance and insight throughout the research. Also I wish to express my deepest gratitude and special thanks to Assoc. Prof. Dr. Erkan Mumcuoğlu for his invaluable support and encouragement.

The work in this thesis is a part of a project named as DİHES (Diş Hekimliği Eğitim Simülatörü), which is supported by The Scientific and Technological Research Council of Turkey. I wish to to extend my sincere thanks to my colleagues, Varlık Kılıç and Anas Abidi, whom I have worked together in this project. Also thanks go to Dr.Dt. Ercüment Önder from METU Health Center, Utku Dede from Ankara University and Assoc.Prof Çiğdem Erbuğ for their suggestions and remarks for the study.

Finally I would like to thank to my friends Varlık Kılıç, Anas Abidi and Süleyman Bideci for their in-valuable supports and friendships. Also thanks go to the other friends Emre, Onur, Faruk, Orçun and Görkem in the lab.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Haptic technology, which enables the user to interact and modify the virtual objects by providing force feedback, has developed rapidly in the last decade. The basic principle of haptic devices [Appendix A, Figure 1.1] is transferring the position of the tool's tip to computer, and supplying 3DOF or 6DOF force depending on the process handled by computer. Trying to reach higher level of reality, haptic devices are integrated with frameworks composed of semi-transparent mirrors, stereo glasses as illustrated in Figure 1.2. The 3-D image of virtual objects are aimed to be registered by the user's hand and haptic device in the workspace of the user by reflecting the inverse rendered image on the screen to the semi-transparent mirror. Although the simulations developed on these frameworks have been being used in several industries including automotive, confectionary, film/video, footwear, games, jewelry, molecular modeling, sporting goods; its impact in the medical field is the most effective one. Hence haptic integrated simulations are threatening the classical methods used in education in medical field.

In most of the medical fields the apprenticeship models; first see, then do, and then teach, have been used as a traditional method for the training of surgeons. However the usage of simulation-based systems, coupled with haptic technology, has become widespread in the medical field. After force feedback has been

adapted in virtual simulations by the widely extended usage of haptic devices, the surgery simulations has provided various features which can not be achieved by the current classical training techniques. Virtual environments have reached a point that their advantages over current training techniques in aspects of reality, assessing the performance, demonstration and cost-effectiveness can not be discussed.

Figure 1.1 Sensable's Phantom Desktop Device.

Figure 1.2 SenseGraphic's 3D-IW Framework.

Plastic models are used for training in most medical areas. In dental training artificial teeth and jaw models are used which can not provide the level of detail and material properties of real tooth [1]. Since repetitive usage of the artificial models is not possible, the artificial teeth usage in training is not cost effective. Another alternative, the use of removed teeth, meets the above requirement nevertheless its quantity is not enough. However virtual simulation environments provide not only different material properties and level of details but also the ability of creating several challenging scenarios which can be faced through real life.

Besides, by the help of virtual simulation environments the precision of the assessment of the trainee's performance is increased. It is possible to define metrics for a specific surgery in simulations. For instance during bone removal operation, simulations are able to figure out the regions which are removed when they are not visible by the user. Also it is possible to define maximum velocities and forces for some critical anatomic regions and observe whether the user exceeded these thresholds. While these metrics are used for a written report, it is also possible to obtain a visual feedback which can show the bone regions in different colors according to the performance of the user for the tasks on that region.

It is also possible for an instructor and a trainee to share the same virtual environment simultaneously. Forces calculated by the interaction of one haptic device with the model can be fed to another haptic device at the same time. By this feature, instructor is both able to experience the interaction forces of the trainee and able to demonstrate how the correct style should be.

Considering the superiority of virtual simulations over the traditional training methods, the transition in medical education is commonly expected. Especially in dentistry education, several dental faculties have started using simulations instead of classical methods like training on plastic models. In other words dental training field is the area which the improvement that haptic integrated simulations bring

has been practically most effective in. The purpose of this thesis is to devise and develop a dental simulation environment in which the user is able to diagnose the decay by probing and then carve the decayed region by suitable dental tools.

1.1 SURVEY OF LITERATURE

Several studies exist in the dental training area, some of which are presented below,

In several dental training programs, Columbia Dentoform Models [2] are used to train students by simulating conditions they will encounter in practice. These are phantom models (Figure 1.3) which can be adapted to bench and chair manikins. It is also possible to design and produce models with different occlusal relationships. Though these models are able to provide different scenarios, the renewal of the jaw models after every use is quite expensive.

Figure 1.3 Columbia Dentoform Models.

The DentSim [3], which is developed by the company DenX to be used for dental training, provides the ability to practice on plastic jaw and face models instead of a virtual simulation environment. But the movements of the student's hands are observed and transferred to a virtual environment by an optical system. A camera which is used to capture the signals emitted from the LEDs (Light Emitting Diodes) located on the dental tool and the patient's jaw model is placed over the

workspace. The location and orientation of the dental tool with respect to jaw is calculated by using signals from the LEDs and displayed as three dimensional tooth images.

Virtual Dental Simulation Environment (VRDTS), which is developed by Harvard School of Medicine by the support of the company NOVINT, is another study on the subject that has a European patent [4]. The project provides the user the capability of working on a decayed tooth, probing the tooth for diagnosis, drilling the decayed region, filling a cavity with amalgam and carving the amalgam to match the original tooth contour. A phantom device is used for haptic feedback and LCD stereo glasses or a computer screen is used for the graphical display of the system. Both the dental tool and the tooth are represented using volumetric representations. Dental tool is composed of two regions: modification and handle regions. Modification region has the ability to drill the decayed part of the tooth while the handle region does not have ability to modify the tooth data. Sample points called as “feeling points” are placed on the surface of the dental tool to handle the interaction of the tool with the tooth. The surface of the tooth data is determined and specified by using an iso-value. The regions with a value lower than the iso-value are considered as outside the surface and the regions with a value higher than the iso-value are considered as inside of the tooth. Like dental tool, the tooth data is also composed of different layers (dentin, enamel, decayed region). The layer and material properties of each voxel are stored in an attribute grid which is the same size with the tooth data. The interaction between the tooth data and the dental tool is handled by checking the position of each “feeling point” with respect to the tooth surface and calculating a force for each of them. The force fed back to the haptic device is the sum of the forces of each “feeling point”.

In another project [5] which is developed in University of Iowa, a virtual dental simulator which proposes to achieve force feedback measured and recorded by experiments before. Instead of diagnosing the cavities by probing, supplying the force signals depending on various sizes and types of gaps in the tooth crown experienced by the dentists is aimed. The forces are recorded by using a probe that

can detect forces between 0 and 400 grams at the dental tool's tip with a frequency of 50 Hz.

In University of Illinois, another dental simulator [6] is developed which focuses on periodontal probing, the use of periodontal explorers in the detection of sub gingival calculus and a variety of other sub gingival topographies.

In Stanford BioRobotics lab [7], a visuohaptic simulation of bone surgery for training and evaluation [8] is devised for general purposes which include bone manipulation. Hybrid data structure is used to represent the bone; a volumetric array stores the density values and attributes of the data while a surface triangles list is used to render the bone graphically. The volume data and surface triangles are obtained from a CT or MR data after a preprocessing procedure [9]. The tool is represented as sample points the drilling power of each is pre-calculated by considering its distance from the central axis of the tool. The force is evaluated by adding the unit vectors of the sample points which immerse the bone volume towards the center of the spherical tool. The simulation also proposes the ability of evaluating the trainee's performance by using predefined metrics and providing a visual and written feedback. Recording performances of surgeons or novices and replaying is possible in order to define metrics [10] special to any operation. The metrics used for mastoidectomy, a surgical procedure in which a portion of the temporal bone is drilled away in order to access inner ear, include the maximum force and velocity of the tool near vulnerable tissues, the probability of each voxel to be removed etc. Moreover the simulator proposes simulating bone dust, providing drilling sounds the frequency of which varies by the applied force, usage of a second haptic device as a suction/ irrigation tool.

Another study [11] proposes to develop a system which is able to simulate probing and cutting operations on tooth. The tooth and tool is modeled by surface triangles. To simulate the interaction between the tool and the tooth, a virtual tool, physical tool and a cutting tool is used. The simulated force has a normal and tangential component. Normal component is evaluated by using a spring constant

and the relative position of the physical and virtual tools and the tangential component is evaluated by using a frictional constant and the normal component of the force. In case the applied force exceeds a threshold, the cutting operation becomes active and the surface triangles of the tooth that are immersed by the tool are modified considering the shape of the cutting tool. In the further phases of the study the force model is evaluated by using a damping matrix [12]. The damping matrix is composed of constants which represent several factors that affect the dental forces such as the type and the shape of the tool, the layer of the tooth, the age of the patient etc. These constants are set by measuring the forces experienced by dentists using an experimental setup.

In all the studies above, the main challenges are the representation methods of the tooth and the tool, the collision detection method, modification of data and the evaluation of the force feedback. All of these challenges are related with each other therefore it is not common to say that a method is best for any of these challenges. Currently, most of the projects have the ability to provide a real time simulation environment while they are still being developed to achieve more realistic simulations and assessment of performance of the user which is a critical issue for vocational education.

1.2 OBJECTIVES OF THE THESIS

The objective of this thesis is to develop a virtual dental simulation environment integrated with the use of a Phantom haptic device. The basic functions that are aimed are:

- Probing the whole jaw model of the patient to find a decayed tooth,
- Feeling the difference between different layers of the tooth and the decay.
- Carving the decay

The software is designed and developed considering the addition of other dental functions like filling the carved region with amalgam and modifying it to fit to original shape of the tooth for future versions.

1.3 THESIS OUTLINE

The 2nd chapter describes the biomechanical differences between the different layers of tooth and decay. A brief review of the experimental work on measuring the properties of the tooth is given and the solutions applied to reflect these differences in simulations are discussed. Then the possible ways of storing the tooth and the tool data are explained in the 3rd chapter. The pre-processing steps of tooth data and jaw model for optimizations are described. 4th chapter illustrates different graphical rendering methods used. The functionalities and performances of these methods are compared. 5th chapter reviews the common methods of haptic rendering in the literature and describes the one used in the thesis. The drawbacks, limitations and the reality of the methods are discussed. Then the 3D stereo effect and the sounds used to achieve more realistic simulations are illustrated in the 6th chapter. The 7th chapter describes both the design and implementation of the software packages done and the properties of the libraries used in the thesis. 8th chapter concludes this thesis by summarizing the work and discussing future directions.

CHAPTER 2

BIOMECHANICAL MODEL OF THE TOOTH

The reality of a simulation environment depends on many factors. These factors depend on the actual purpose of the simulator. If the simulation is to be graphical with feedback, the quality of the 3D models in the environment, the reality of the feedback forces affect the simulation directly. Since one of the aims of this thesis is to achieve a simulation of a highly detailed 3D model structure of a tooth with decay where upon interaction the user shall get a feedback in the form of a touch or a grinding sensation, a survey is required in order to develop an understanding on how to perform these tasks.

In this chapter we will discuss: the biomechanical properties, data and the structure of the tooth to be simulated, the model used to evaluate the forces for the feedback.

2.1 BIOMECHANICAL ASPECTS OF TOUCHING AND GRINDING

The touching and grinding forces are dependent on some mechanical properties of the material that is touched and grinded. In order to understand these properties a comprehensive literature survey is done.

Modulus of elasticity and structural damping coefficient are the properties that affect the force during touching. Modulus of elasticity represents the amount of

stress that should be applied to create one unit of strain in a material. It is independent of material's geometry, while rigidity constant is calculated using length, cross-section area and modulus of elasticity. Structural damping coefficient is the frictional force of material against the unit elongation speed. One can evaluate the motion of an object under a dynamic force by using these two coefficients, density and geometry of the material.

On the other hand specific cutting energy is the property that affects the forces during the grinding operation. This constant is equal to the energy needed to remove 1 m^3 of material by using 1mm chip thickness. The relationship between this coefficient and the thickness of the removed chip is established by empirical studies. The grinding force can be calculated by using volumetric chip removal speed, the chip thickness and the specific cutting energy. On the other hand for cutting events which the exact geometry of the tool is known, the forces can be evaluated by using the shear and tensile strengths of the material[13].

One can conclude that the main mechanical properties which affect touching and grinding operations are structural damping coefficient, elasticity of modulus, specific cutting energy. In the specific case of probing and grinding the tooth, no experimental studies are found related with the structural damping coefficient and specific cutting energy. However there are several studies related with the hardness and elasticity of modulus of tooth. These studies are briefly summarized after the structure of the tooth is explained.

2.2 STRUCTURE OF THE TOOTH

There are two main layers which form the bone structure of the tooth; dentine and enamel. In the middle part of dentine there is a pulp region which contains blood vessels and nerves. Enamel, which protects the underlying dentine, is the hardest substance in the human body, harder than the bone. It gains its hardness from tightly packed rows of calcium and phosphorus crystals within a protein matrix structure. Dentine, the major component of the inner part of the tooth, is slightly softer than

enamel, with a structure more like bone. It is elastic and compressible in contrast to the brittle nature of enamel. Dentine contains tiny tubules throughout its structure that connect with the central nerve of the tooth within the pulp. Dentine is a `live` tissue compared with enamel. In the middle part of the tooth, pulp forms the central chamber of the tooth. The pulp is made of soft tissue and contains blood vessels to supply nutrients to the tooth, and nerves to enable the tooth to sense heat and cold [14]. The structure of the tooth is illustrated in Figure 2.1.

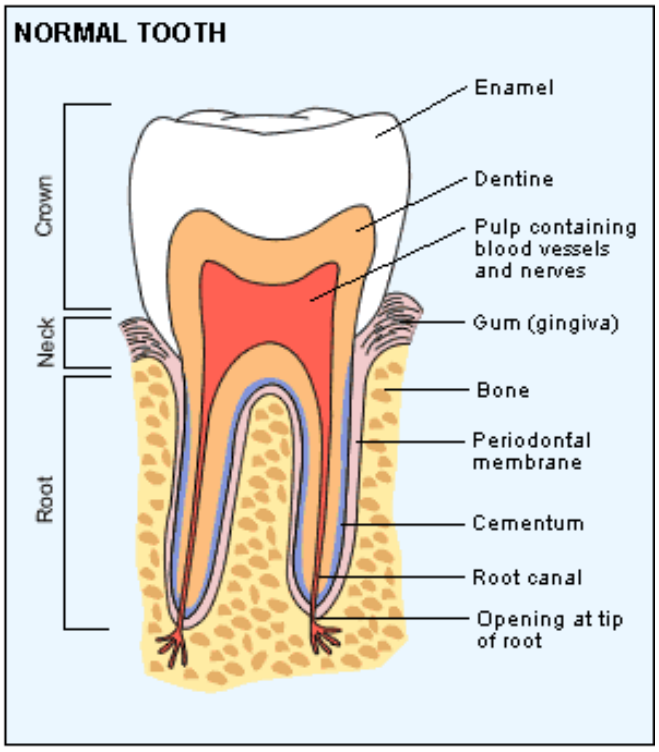


Figure 2.1 Structure of a normal tooth.

In this thesis, the aim is to diagnose a tooth decay by probing and drilling the bone structure in simulation environment. Therefore, the mechanical properties of the pulp are not considered for the simulation. A parametric model is built to demonstrate the biomechanical differences of dentine, enamel and decay.

2.3 SURVEY OF LITERATURE ON BIOMECHANICAL PROPERTIES

After completing the survey on the main factors that affect the general forces for the touching and grinding, we now focus on the biomechanical properties of the tooth. The studies in this field deal with the elasticity of modulus and the hardness of the tooth. These studies mostly use the molars which are the rearmost and most complicated kind of tooth in most mammals. Adult humans have twelve molars, in four groups of three at the back of the mouth as illustrated in Figure 2.2.

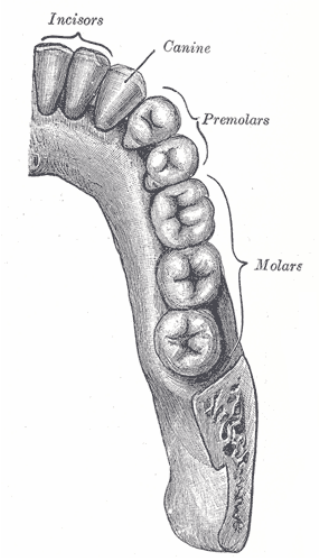


Figure 2.2 The types of teeth in the jaw.

The studies reveal the fact that biomechanical properties of the tooth change with layers of tooth, age and from person to person. Some of these studies compare the properties of different layers while some deal with the change throughout a layer. These studies are summarized below.

A study [15] uses molar teeth to measure the hardness and elasticity of modulus in the dentin layer. Eight primary molar teeth are axially sectioned, embedded in resin and fine polished. Then linear arrays of indentations are done on coronal dentine, from the pulp wall to dentine-enamel junction parallel to the tubule direction under a force load of 25mN. Each indent is accomplished by a cycle of 25 incremental

loading steps with a delay of 0.1 s at every increment, a 30s delay at the maximum load (25mN) followed by another 25 increments for unloading. The hold period at the maximum load enabled creep about the contact site to ensure minimal creep during unloading and thereby generate a reliable elastic modulus value from the slope of tangent to the unloading curve. The software associated with the Ultra Micro Indentation System (UMIS – 2000, CSIRO Australia) [16] analyses the data and calculates the elasticity of modulus from the load-displacement curve. The results reveal that the hardness and elasticity of modulus are not constant through the dentin layer; their values increase from pulp to enamel. Elasticity of modulus and hardness values measured in the region closest to pulp in the dentine, middle dentine, the transition between dentine and enamel (DEJ) are listed in Table 2-1.

Table 2.1 Changing hardness and elasticity of modulus coefficients through layers of tooth.

	Hardness(GPa) mean ± SD	Elasticity of Modulus(GPa) mean ± SD
500μm to pulp	0.52 ± 0.24	11.59 ± 3.95
Middle Dentine	0.85 ± 0.19	17.06 ± 3.09
300μm to DEJ	0.91 ± 0.15	16.91 ± 3.85

Another study [17] follows a similar method to measure the hardness and elasticity of modulus of layers of tooth, but it considers the dentine and enamel layers as homogenous. Similarly primary molar teeth are enclosed in a rubber mould and set in a cold curing epoxy resin. Once the resin has set, the embedded teeth are removed from their moulds and the crown of each tooth is sectioned half way between the tip of the buccal cusp and the cemento-enamel junction. The finished tooth specimens are mounted on metal basis with wax. Then thirty indentations are made in enamel and dentine using a Berkovitch indenter, 15 of which are subject to a load of 50mN and 15 to a load of 150mN. The UMIS system converts the force/penetration graph for each indentation into hardness vs. depth graph from which values for mean hardness and elastic modulus are calculated. The average values of the layers obtained in this work are listed in Table 2-2.

Table 2.2 The hardness and elasticity of modulus coefficients for dentine and enamel.

	Hardness(GPa)	Elasticity of Modulus(GPa)
Dentine	0.92 ± 0.11	19.89 ± 1.92
Enamel	4.88 ± 0.35	80.35 ± 7.71

The ultimate tensile strength, the peak stress on the engineering stress-strain curve, is measured through different layers of tooth in [18]. Ultimate tensile strength represents the amount of tensile strength that a material can be subjected to before failure. Although this property of tooth does not have a direct effect on grinding, the difference in tensile strengths of layers of tooth may be considerable in other functionalities of simulation. In this study, twenty human third molars are cleaned of gross debris and placed in distilled water for 24 hours before beginning the experiment. The roots of the teeth are removed approximately 3mm below the cemento-enamel junction using a diamond disk. Then slices of approximately 0.7 mm thickness are obtained by vertically and serially sectioning the remaining crown. Each specimen is fixed to grips of the micro tensile testing device with cyanoacrylate glue and tested in tension at 0.5mm/min in a universal testing machine (4411, Instron Co., Canton, MA, USA) until failure. After fracture, the specimen is removed from the testing apparatus and the cross sectioned area at the site of the fracture is measured with a digital caliper. The mean tensile strength values of the layers are expressed in MPa. The experimental results are listed in Table 2.3. It is observed that the ultimate tensile strength has an anisotropic characteristic through enamel layer.

In order to model the treatment of decay, a literature survey on the theory of grinding operation is accomplished. There exist studies [13] on grinding operation which propose simulating the relationship between the force and the chip removal rate. These studies address grinding operation where the results are confirmed by experiments on metal grinding and the models have a high computational workload. Therefore the information obtained from these studies is not suitable for tooth simulation software.

Table 2.3 The ultimate tensile strength (MPa) of tooth structures.

	Mean(SD)
Superficial dentine	61.6(16.3)
Middle dentine	48.7(16.7)
Deep dentine	33.9(8.0)
Dentin – enamel junction	46.9(13.7)
Parallel enamel	42.2(12.0)
Transversal enamel	11.5(4.7)

There are studies which implement the grinding operation by using haptic applications [12, 19]. In [19], a mathematical model which includes the diameter, rotation speed, active contact area with material, velocity, cutting depth of the grinding tool as main parameters is established. By using this mathematical model, which is statistically tested by experiments, a relation between the above factors and force is established. In addition to these factors the layer of the tooth and the type of the tool are also considered in a similar study [12]. Likely, this study also creates a mathematical model to simulate the grinding operation on tooth. The haptic algorithms used in these studies are discussed in Chapter 5.

2.4 BIOMECHANICAL MODEL USED

After a comprehensive literature survey about the layers of tooth, the biomechanical differences between these layers and the effect of these differences on the haptic simulations, a parametric model is designed.

First the tooth is segmented into its layers, decay is located in it, then the parameters for each layer and the decay is set by the help of dentists to reach a higher degree of realism for the simulation.

The tooth data used in the simulation is a voxel array of dimensions 128x128x80. Each voxel has a grayscale value between 0 and 255. The lower grayscale values around 0 represent the free space and the noise, while the higher grayscale values

represent the hard tissue. The enamel layer of the tooth is harder than the dentine layer therefore it has higher grayscale values than enamel has. In order to demonstrate the difference between two layers, the tooth bone structure is segmented.

For the segmentation of dentine and enamel layers, first a threshold method is applied to use the grayscale value difference between the two layers. Because of the outer layer's low grayscale value on the enamel layer a correct threshold can not be adjusted. In other words if a high threshold is applied to distinguish between two layers, the enamel layer becomes smaller than it should be. In the case of using a low threshold value the labeled enamel region extends towards the lower part of the tooth which is actually dentine.

It is assumed that the boundary between dentine and enamel is horizontally flat; the upper part of the tooth is segmented as enamel and the lower part as dentine. Although this assumption seems to eliminate the detailed boundary between two layers graphically, its effect is not so critical in the haptic rendering. This is because of the low resolution of the tooth data and the small image size of the tooth on the screen.

After the segmentation of the layers, decay is created in a specified location as a spherical region using the volumetric tooth data. This spherical region is located near to the upper surface of the tooth. Therefore, the decay has geometry of an intersecting part of a sphere with the shape of the tooth as illustrated in Figure 2.3. The decayed region is darker than the original color of the tooth and the mechanical properties of the decay are set by a dentist to achieve a feeling similar to the real life experiences of the dentists.

The difference in mechanical properties of dentine and enamel is illustrated by considering the hardness and elasticity of modulus values of the layers. The haptic rendering algorithm applied uses stiffness, damping, static and dynamic friction coefficients. A direct proportion can be established [Appendix B] between elasticity

of modulus and the stiffness. Therefore the average ratio of elasticity of modulus coefficients of the layers can be applied to stiffness coefficients as well. The ratio of approximately 5, as observed in [17], is set as a default stiffness ratio for dentine and enamel layers.

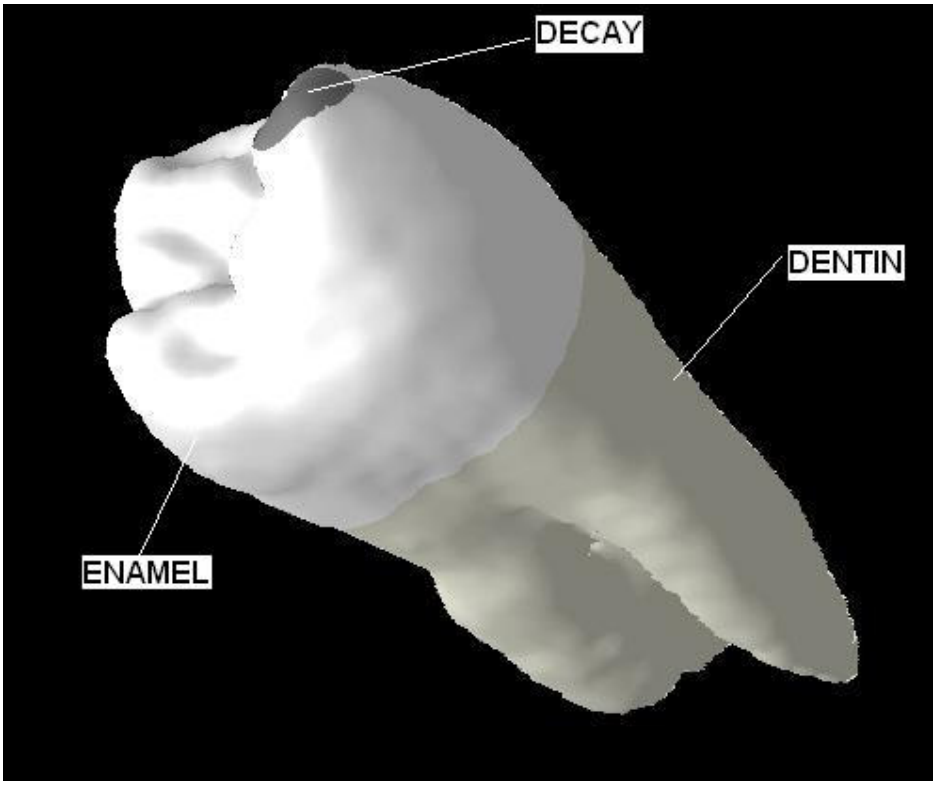


Figure 2.3 A screenshot of segmented tooth and decay from the software.

In order to demonstrate the surveyed biomechanical differences between the layers of tooth, decay and palate, a parametric model is implemented. The haptic algorithm developed in this study allows the user to change the stiffness, damping, static and dynamic friction coefficients in real time by using the interface illustrated in Figure 2.4. By this way the parameters can be adjusted for palate, dentine, enamel and decay separately. After the recognition of the structure (palate, dentine, enamel, decay) which the user touches using the haptic device, the parameters that are used to evaluate the force are adjusted to the properties of that structure.

The stiffness of the structure represents the resistance of the elastic body to deflection or deformation by an applied force. Stiffness is used as a spring constant in the spring model used for the haptic rendering. The haptic algorithm uses the damping coefficient of the structures as the resistance to velocity in the spring model. Static and dynamic friction coefficients determine the roughness of a surface. Static friction represents the resistance to the start of motion on the surface, while dynamic friction represents the resistance to the movement on the surface.

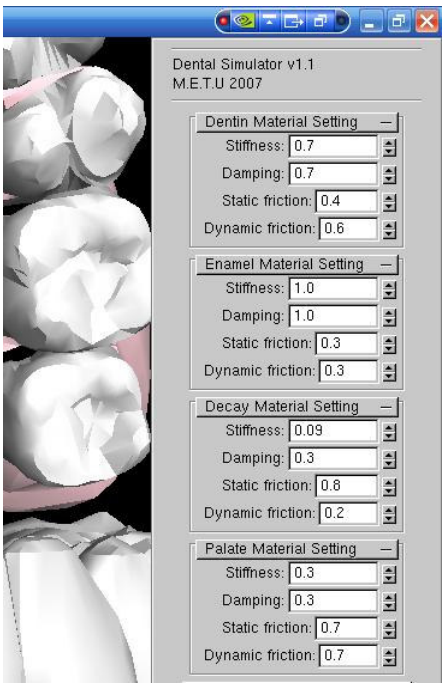


Figure 2.4 A screenshot from user interface including the mechanical properties of different structures.

It is possible to change the properties of the structures by using the user interface of the software as illustrated in Figure 2.4. Using the user interface one can adjust the mechanic properties of the structures while experiencing the force by the haptic device. First the values of biomechanical properties of the structures are set to the default values which give roughly realistic feeling. Then the parameters of decay, dentine, enamel and palate are tuned by a dentist to achieve a realistic feeling similar to real experiences of experts in the field. The tuned parameters are shown in Table 2.4.

Table 2.4 The parameters used in haptic algorithm for different tooth structures.

	Palate	Dentine	Enamel	Decay
Stiffness	0.3	0.7	1.0	0.09
Damping	0.3	0.7	1.0	0.3
Static Friction	0.7	0.4	0.3	0.8
Dynamic Friction	0.7	0.6	0.3	0.2

CHAPTER 3

DATA REPRESENTATION

In this chapter the models used to represent the deformable tooth, complete jaw model and dental tools will be introduced. The effects of the data representation type in the simulation will be discussed.

The data representation of the models in simulations is dependent on the graphical and haptic rendering algorithms used in the simulations. There are two types of representations: volumetric and surface data. It is not possible to claim which one is better than the other because each one has some superiority over the other. The representation of the models depends on the purpose of the simulation. The way the models are represented also affects the overall performance of the software. A volumetric model is preferred in a simulation that requires rendering the inner structure of the models. Whereas, the usage of surface model for graphical rendering has better speed performances. Each of the methods is commonly used in the literature. Both tool and tooth are represented volumetrically in [4]. While a hybrid representation is used in the study [9] which stores the tool data volumetrically and keeps the bone structure in a list of surface triangles. On the other hand [9] keeps the tool and the tooth as a list of surface triangles.

3.1 DEFORMABLE TOOTH

The tooth which can be modified is volumetrically stored in a 3D voxel array. This allows the assignment of properties for each voxel and allows constant time access to voxels colliding with or occupied by the tool. Each voxel has a scalar density value and a gradient vector which represents the magnitude and direction of change in the density value. Different graphical rendering methods use the density and gradient information from this voxel array to create the data structures for graphical representations. The surface based method implemented on CPU, evaluates the surface mesh which is made of triangles by applying the marching cubes [20] algorithm on the 3D data. Also the voxel array is used as a 3D texture in the GPU for the ray-casting graphical rendering methods implemented in GPU.

The tooth data used in the thesis is a voxel array of dimensions of 256x256x161 which is obtained from a demo program [21]. It is possible to load any size of 3D data in the simulation. However, it is observed that when the size of the 3D voxel array increases the speed performance of the simulation decreases. Currently the simulation works with an array of dimensions of 128x128x80 which is obtained by down-sampling the original data.

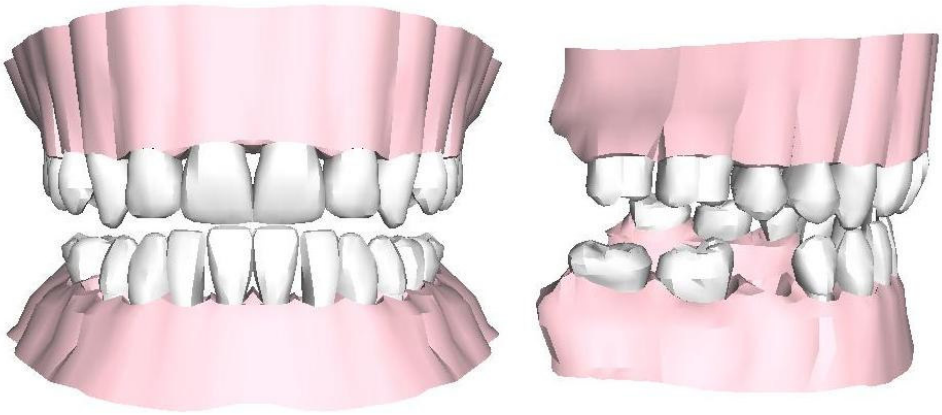
3.2 JAW MODEL

After modeling the deformable volumetric tooth data which has decay in it, a complete jaw model is incorporated in the simulation to give higher level of realism. Unlike the deformable tooth the teeth and jaw models are stored as a surface data not to negatively influence the simulation speed performance and haptic rendering stability. Besides storing the entire jaw model volumetrically with the same resolution of the deformable tooth will require too much memory which current computers cannot support. The jaw model can not be modified, however it will allow collision detection in case of any contact with the haptic device. The jaw model consists of

- The upper jaw

- The teeth model for the upper jaw
- The lower jaw
- The teeth model for the lower jaw

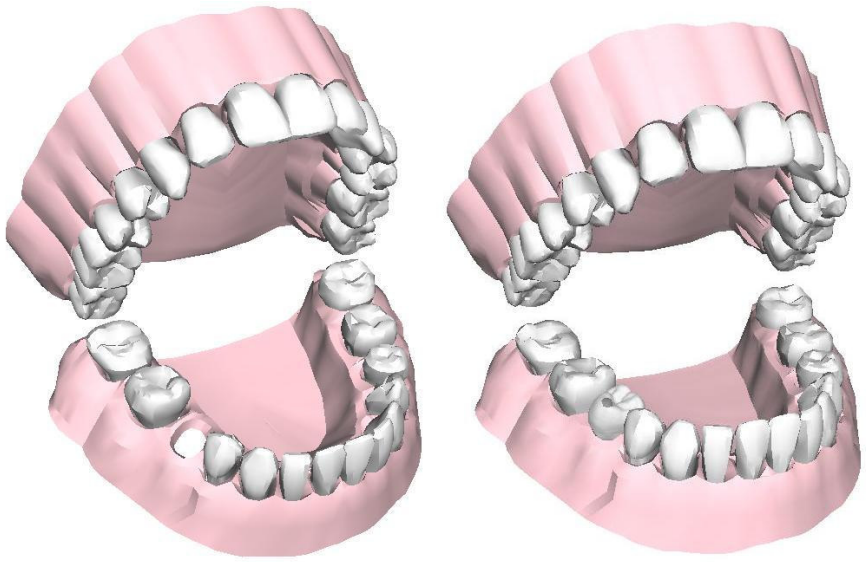
The upper jaw, lower jaw and their respective teeth set (14 teeth for each set) are modeled and designed in CATIA® and then converted to Wavefront object model format. The loading of the model into the simulation environment is handled by using the 'glm' library [22]. The 'glm' library is basically a parser and reader which understands and supports the data included in "obj" files. This data after being parsed is represented as a 'glm' model. The 'glm' library also includes many useful functions to implement and manipulate the 'glm' models, for example: drawing (to draw the model in OpenGL), scaling, smoothing, etc...



(a) (b)

Figure 3.1 The jaw Model without deformable tooth. (a) Front View, (b) Side View.

After loading and parsing the entire jaw model, the deformable tooth is scaled and translated to fit the place of the missing tooth which is not loaded as illustrated in Figure 3.2(b). The mechanical parameters that affect the haptic force are set separately for palate, decay and the teeth as described in Chapter 2.



(a) (b)

Figure 3.2 Different Views of Jaw Model. (a) The jaw model fully opened to allow the dentist to work on it. (b) The deformable tooth is scaled and translated to fit into the jaw model.

3.3 DENTAL TOOLS

There are two types of dental tools: The tools that are used to: modify the tooth, and to contact the tooth. The contact tools are used to diagnose the decay and these cannot modify the tooth data. The modifying tools are used to touch and modify the tooth data. For the tools there are two main criteria: graphical representation and haptic representation of the tools which are discussed in the proceeding sections.

3.3.1 Graphical Representation

The tools used in the simulation are represented as surface data for graphical rendering. Different types of dental tools are modeled in CATIA® and then converted to Wavefront object model format. The loading of the model into the simulation environment is handled similar to jaw model by using the 'glm' library. The position and orientation of the tools are synchronized with the position and orientation of the 6 DOF haptic device. Some dental tool models are illustrated in Figure 3.3.

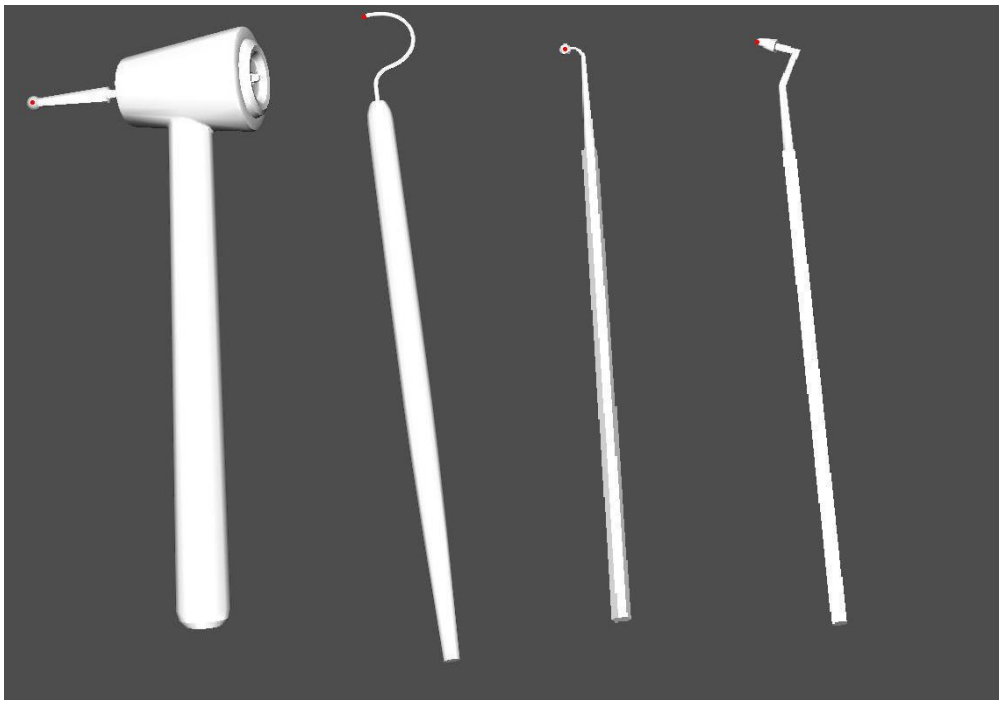


Figure 3.3 Different tool models. The red points represent the tool tips used in haptic rendering.

3.3.2 Haptic Representation

The dental tools have a point on their tips for the touching operation. The haptic rendering algorithm assumes the tool to be a point whose position changes with the movement of the haptic device. The haptic rendering algorithm is implemented by using OpenHaptics-HLAPI library which uses the "god-object" method discussed in Chapter 5. The algorithm evaluates the forces assuming that the tool is defined as a point in space. The point of the tools which is used in evaluating the haptic forces is located on the tip of the tools that the dentists use to examine tooth structures. The tip points of the tools are marked as red in Figure 3.3.

For the modification of the tooth data used in treatment of decay, using a point representation is not sufficient. The modifying tool has an active modifying region which modifies the tooth data in case of collision with the tooth. This region is represented volumetrically, and in case of modification each voxel of this region is checked for a collision with corresponding tooth voxels. The voxels of active

modifying region is stored in a hash table. Each tool voxel in the hash table has positions relative to the tool center. The 2D representation is illustrated in Figure 3.4. If a 3D array was used to represent the active modifying region, all of the voxels in the array would be checked for collision detection. For instance to represent a spherical region, a 3D bounding box array which includes the sphere should be used. The voxels which are not in the sphere but are in the bounding box would be unnecessarily checked for collision detection. By storing the coordinates of the voxels of modifying region in hash-map, unnecessary traversal of tool voxels is prevented. In the simulation a spherical region is used for modification and only the voxels that are in the sphere are stored in the hash-map.

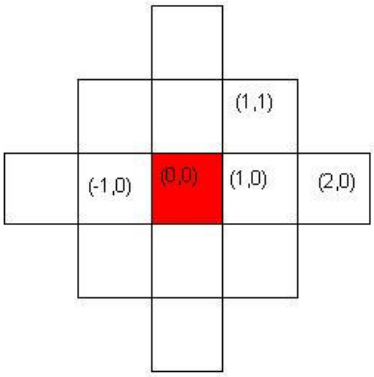


Figure 3.4 2D representation of the active modifying region of the tool. The red block represents the tool tip. The relative coordinates of each modifying voxel are stored in the hash map.

This representation of the modifying region allows the usage of any shape for the modifying region. Any regular or irregular shape can be volumetrically defined and then the voxel coordinates with respect to the origin of the tool can be added to the hash map. The modification algorithm using this data structure is described in Chapter 5.

CHAPTER 4

GRAPHICAL RENDERING

Quality and performance of the graphical rendering in a simulation are the most critical issues that affect the realism of the simulation. Achieving a realistic, high quality image on the screen is not sufficient. The frame rate of the simulation should be above a threshold value in real time applications. The human eye recognizes the animations continuously if the refresh rate is over 30 frames per second. To increase the quality of the image, the algorithms include complex operations like per pixel color and lighting calculations. However, these complex operations negatively influence the refresh rate of the image. Therefore, there should be a compromise between the refresh rate and the quality of the image in a real time virtual application. In this chapter the algorithms implemented to render the deformable tooth and jaw model will be discussed and performances of different rendering algorithms will be compared.

The thesis proposes graphically rendering an entire jaw model, dental tool and a deformable volumetric tooth model. The jaw model and dental tools are stored and graphically rendered as surface data whereas the deformable tooth is kept in a 3D voxel array. Different graphical rendering approaches are implemented to render the deformable tooth to compare speed performances and quality of the rendering. First approach is the marching cubes algorithm [20] implemented on CPU. The second approach is ray casting [24, 25], which is implemented on GPU (Graphical

Processing Unit). The second approach includes iso-surface and volume rendering techniques by ray casting.

4.1 DENTAL TOOLS AND JAW MODEL

As mentioned in Chapter 3, the jaw model and the dental tools are represented as surface data. The surface triangles are kept in a glm model [22] format and compiled into an OpenGL display list for efficiency. The graphical rendering of these display lists is handled by using OpenGL library.

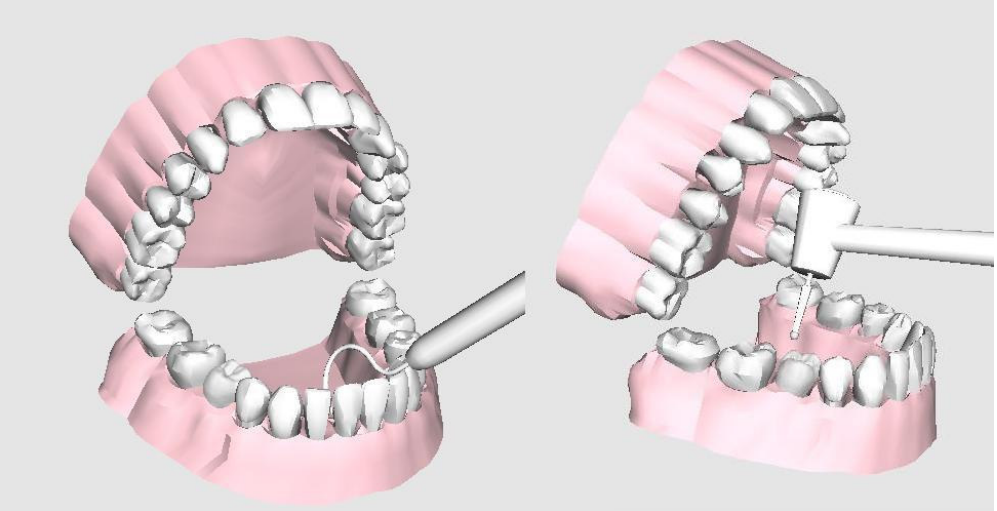


Figure 4.1 Two screenshots showing the entire jaw model with different tool models.

The jaw model is composed of an upper and a lower jaw. In the graphical loop, the lower jaw model is rendered first at the origin of the world coordinates. Then a rotation around the X axis is applied to the upper jaw model to simulate the opening and closing operation of the mouth. The angle of opening the mouth can be changed from keyboard in real time. It is also possible to change the view point, scale and translate the model by using the mouse. The dental tool's tip (described in Chapter3) is considered as the origin of the tool and the tool is rendered such that the position of the tip is the same as the position of the haptic device. Then the orientation of the tool is made the same as the orientation of the haptic device. The images of the jaw model together with the dental tool can be seen in Figure 4.1.

4.2 DEFORMABLE TOOTH

The deformable tooth is stored as a 3D voxel array with dimensions of 128x128x80. Different graphical rendering techniques, surface and volumetric, are used to render the deformable tooth. The marching cubes algorithm [20], iso-surface and volumetric rendering by ray-casting on GPU are the graphical rendering methods integrated to the simulation. In order to examine the inner structure of the tooth by changing the transparency and color values, a transparency mode is implemented in the GPU. A comparison of the methods is discussed at the end of the chapter.

To increase the speed performance, optimization steps are integrated during the loading of data or graphical loop. In ray-casting methods a pre-processing step, an operation that creates a minimum fitting bounding box that covers the tooth inside the 3D voxel array, is integrated into the software during loading operation. During the modification of the tooth data the graphical data structures are locally updated instead of updating the whole data to achieve acceptable frame rates for a real time simulation.

4.2.1 Iso-Surface Rendering on CPU (Marching Cubes)

The iso-surface based rendering technique is achieved by the well-known marching cubes [20] algorithm. The marching cubes algorithm is given an iso-value as an input. The algorithm evaluates the triangles that form the surface corresponding to the iso-value by using interpolation between the neighborhood voxels. At the beginning of the program the algorithm is run for all of the voxels in the volumetric data. Each voxel's grayscale value lies between 0 and 1 and the iso-value used by marching cubes algorithm is 0.5. The normal vectors of the vertices of the triangles are calculated by trilinear interpolation using the gradient component of the tooth data.

The triangles created by the marching cubes algorithm are stored in a hash map named as *Surface Display List* with an index generated using the 3D coordinates of

the corresponding tooth voxel the triangle belongs to. During the graphical loop all the triangles in the *Surface Display List* are rendered. The tooth model rendered by this method is depicted in Figure 4.2.

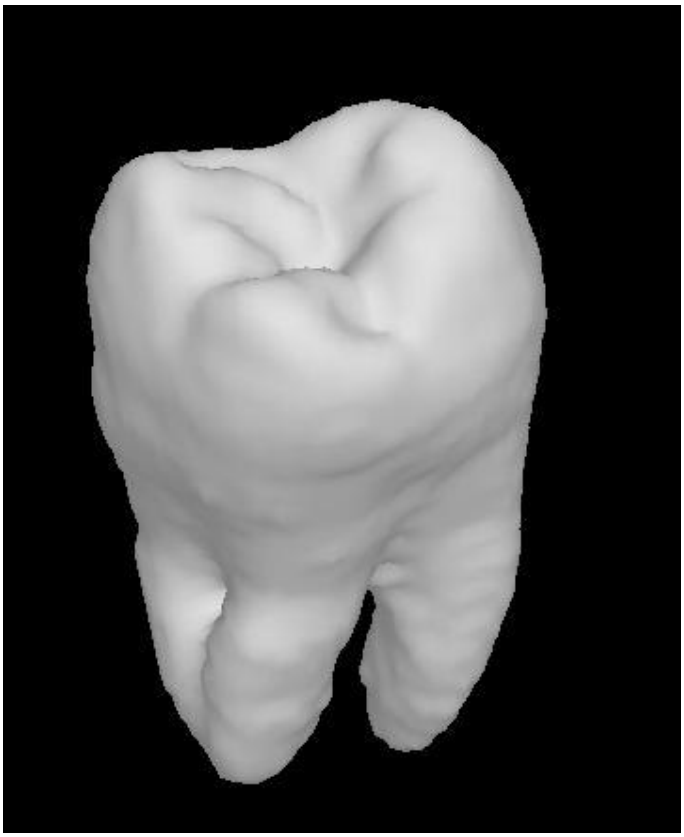


Figure 4.2 The deformable tooth rendered on CPU by marching cubes method.

The hash map structure allows the local update of the surface data during modification. The voxels which expose a modification are added to a list named as *Modified Voxels List*. The triangles belonging to the voxels in the *Modified Voxel List* can be easily found by using the index showing the voxel number in the *Surface Display List*. All of the triangles belonging to the voxels in the *Modified Voxel List* are deleted and new triangles for these voxels are re-calculated from the modified tooth data by marching cubes algorithm. Finally, new triangles are added to the *Surface Display List* which is continuously rendered in the graphical loop. The modification algorithm is explained in a detailed way in Chapter 5.

4.2.2 Ray Casting on GPU

Unlike the marching cubes algorithm, the ray casting methods use a 3D texture instead of triangles to render the scene. 3D array keeping the volumetric tooth data is used to bind the texture. Then the texture is stored in the graphical card. In the display loop, a bounding box which covers the 3D texture is drawn by activating the algorithm developed in the fragment program. The algorithm in the fragment program is run for every pixel on the surfaces of the bounding box. In this thesis, a similar algorithm with the ray casting algorithm in [24, 25] is implemented. The per pixel depth value calculation, which allows several textures to be rendered in the same scene, is also adapted to the algorithm.

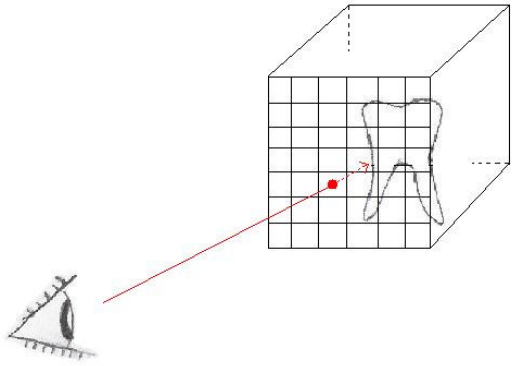


Figure 4.3 The general idea of the ray casting method.

The graphical rendering methods implemented on GPU depend on the idea of sending rays from optic center of the camera through each pixel on the image plane. When the stopping condition is satisfied the ray is stopped. The stopping condition changes depending on the type of the algorithm. The color of the pixel is decided according to the data gathered on the ray's path during traversal. This approach may be utilized to display both surface and volumetric data and it provides better

visualization than the classical approaches in existence of shiny surfaces. The general approach used in ray casting operation is illustrated in Figure 4.3.

The ray casting algorithms provide high quality images since lighting and color calculations are done for each pixel individually. However this computational workload causes low frame rates in the simulations. In order to increase the performance of the ray casting algorithms some optimization processes are integrated to the simulation. These are minimizing the bounding box including the 3D texture and local update of the texture during modification similarly in Iso-surface rendering on CPU.

The size of the bounding box that covers the 3D texture directly affects the frame rate of the ray-casting algorithms, since the ray is cast from the pixels on the surface of the bounding box into the texture. The path the ray travels through the texture becomes longer as the size of the bounding box increases. Besides because of the increased number of pixels that the bounding box has on the screen, the ray casting operation is evaluated for much more number of pixels. It is observed that the smaller the size of the bounding box, the greater the frame rates as expected theoretically.

During the loading of the data to 3D array, minimum and maximum coordinates that has a meaningful grayscale value instead of noise are found for X, Y and Z axis. The minimum and maximum coordinates found by this method [23], form a minimized box that covers the tooth completely in X, Y and Z axis. As a result the empty voxels of data at the edges are eliminated and the size of the 3D texture is reduced, which directly affects the speed. The bounding box with and without preprocessing step is illustrated in Figure 4.4.

As in the surface rendering on CPU, tooth data is updated locally when the user drills it. Instead of updating the surfaces, the 3D texture in the graphics card is locally updated by using ‘glSubtexture’ function of OpenGL. The region where the modification occurs is bounded by a bounding box. Then the values of the 3D

texture in the graphics card are locally updated by using the corresponding updated tooth data. This operation is repeated for every graphical loop during modification.

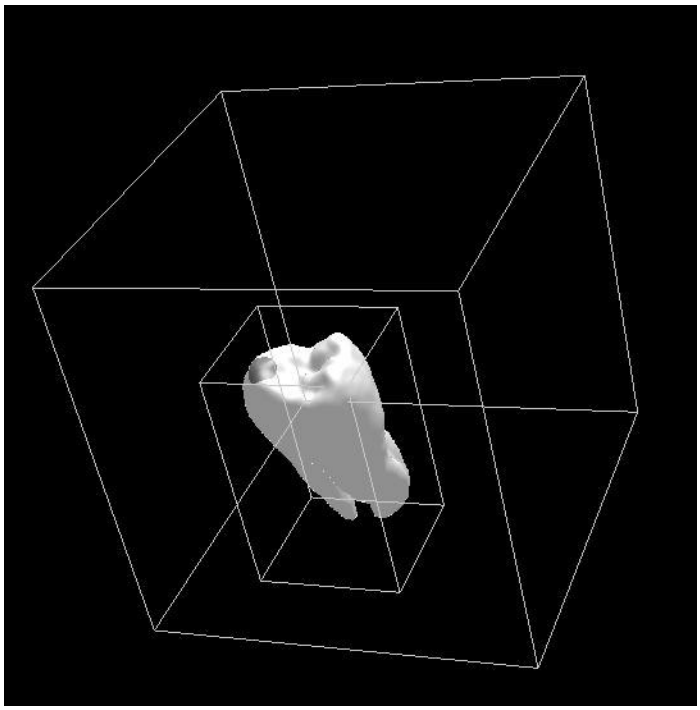


Figure 4.4 Bounding Box Optimization. The small bounding box is the one after preprocessing of the data, while the big box depicts the original size of the bounding box.

In the thesis three distinct fragment programs for the GPU are developed. These fragment programs implement the ray casting method for displaying the volumetric data which is stored in the GPU Texture structure. Two of the fragment programs are used to achieve different rendering methods for the deformable tooth placed in the jaw model; the first fragment program draws iso-surfaces and the second one displays the whole volumetric data by blending. A transparency mode which can show the inner structures of the tooth in different colors and different opacities is also integrated in the simulation by the third fragment program.

4.2.2.1 Iso-surface Rendering by Ray Casting on GPU

The aim of the fragment program is to find the surface corresponding to a given iso-value in the 3D texture. The algorithm runs for every pixel on the surfaces of a

bounding box covering the 3D texture and sets the color and depth values individually. A ray is cast from the corresponding pixel on the bounding box into the 3D texture in the direction from camera to the pixel. The ray casting operation is stopped when the density value stored in the texture just crosses the specified iso-value, and the color of the pixel is set to a specified value by an opacity value of 1. Then according to the gradient information gathered from the texture, Phong lighting model [26] is applied to determine the diffuse and specular components of the light. The depth value of each pixel is set individually according to the z – value of the final ray position which is evaluated by the following equation used in [23];

$$d = 1 - \frac{1/z - 1/F}{1/N - 1/F} \qquad \text{(Equation 1)}$$

Where, d is the depth value between 0(closest) and 1(farthest) and is written to the depth buffer, z is the final position of the ray, F is the far plane distance and N is the near plane distance.

In the case of no iso-value crossing, the color of the pixel is set to black with an alpha value of '0' and the depth is set to the maximum depth value '1'. This provides the ability of displaying another intersecting or non-intersecting structure, volumetric or triangular, in the same scene. This feature enables the rendering of the tooth together with the entire jaw model without any inconsistency.

4.2.2.2 Blending by Ray Casting on GPU

Similar to the preceding fragment program, this mode proposes displaying 3D texture by ray-casting operation. However, instead of finding an iso-surface, a specified color is blended by using the density values of the texture while the ray travels in the volume. Each pixel's color is set as the final blended color instead of drawing iso-surfaces. The depth value of the pixel is corrected according to the z value of the ray when it first faces a non-zero density value by using equation 1. Per pixel lighting calculation is applied by using the gradient value of the texture at the

point where the depth is set. To optimize the speed performance, the ray-casting operation is stopped when the blended alpha value saturates.

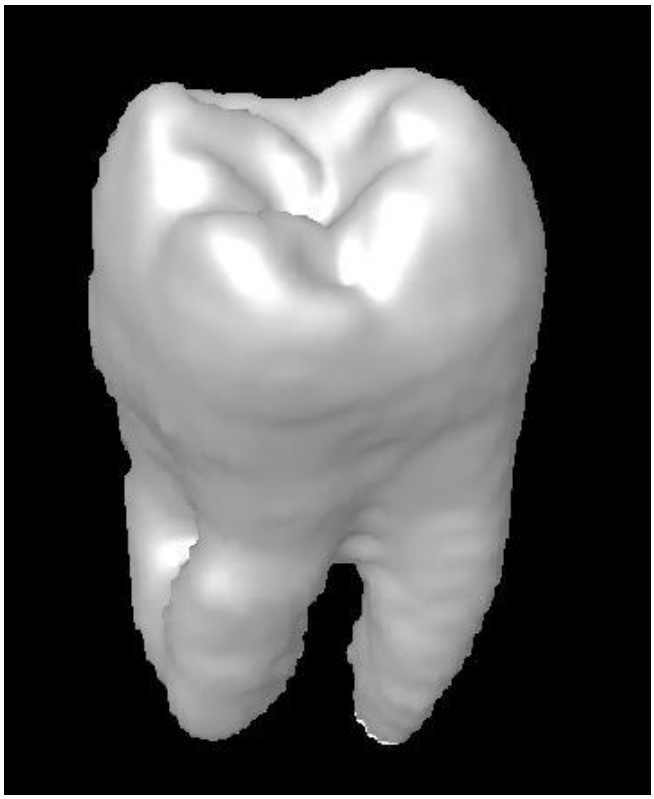


Figure 4.5 The deformable tooth rendered on GPU by using a iso-surface.

4.2.2.3 Transparent Rendering Mode on GPU

In the study [27], a fragment program that enables changing the transparency and RGB color values of the voxels by using a 1D transfer function is implemented. The algorithm used in this fragment program is adapted to a transparency mode in the simulation. In this mode, it is possible to view the histogram of the teeth and change the histogram mapping function in red, green, blue and alpha channels separately. In other words this mode enables the user to obtain a tooth with different colors and transparencies to visualize the inner structure of the tooth.

During the loading of the tooth data, histogram of the image is calculated by applying a threshold to clear noise. Then a histogram mapping function is created

by using the formula y = x for all red, green, blue and alpha channels. This histogram mapping function is stored in a 256x4 size array, including the RGBA values of each color. The array, default values of which will cause a colorless tooth with the original density values, is bind to the graphics card as a 1D texture together with the 3D texture of the tooth.

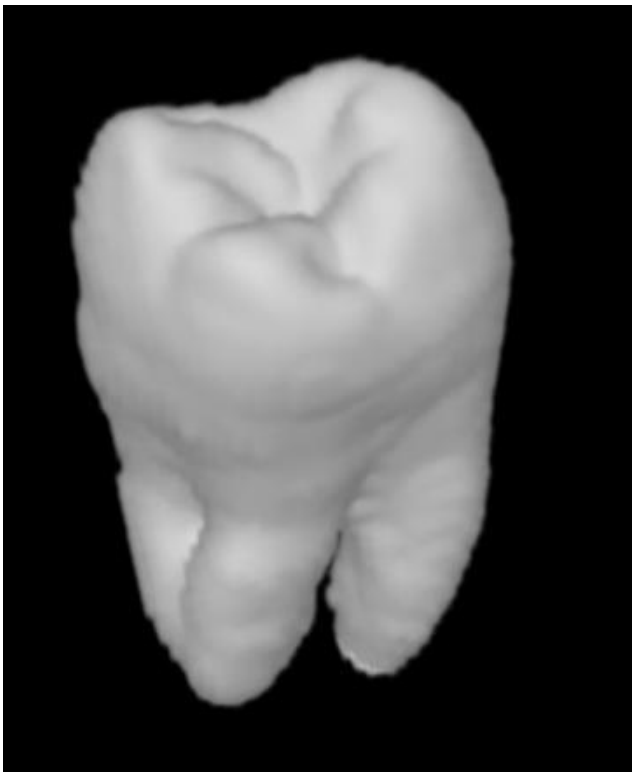


Figure 4.6 The deformable tooth rendered on GPU by using blending.

The algorithm that the fragment program applies is similar with the one used in blending on GPU. A specified color is blended while the ray travels through the texture similarly in the previous algorithm, however the color to be blended is different for each voxel and determined by using the histogram texture. As the ray travels through the texture, the density value of the voxel in the 3D texture is used to specify the color to be blended for that voxel by using the 1D histogram mapping texture.

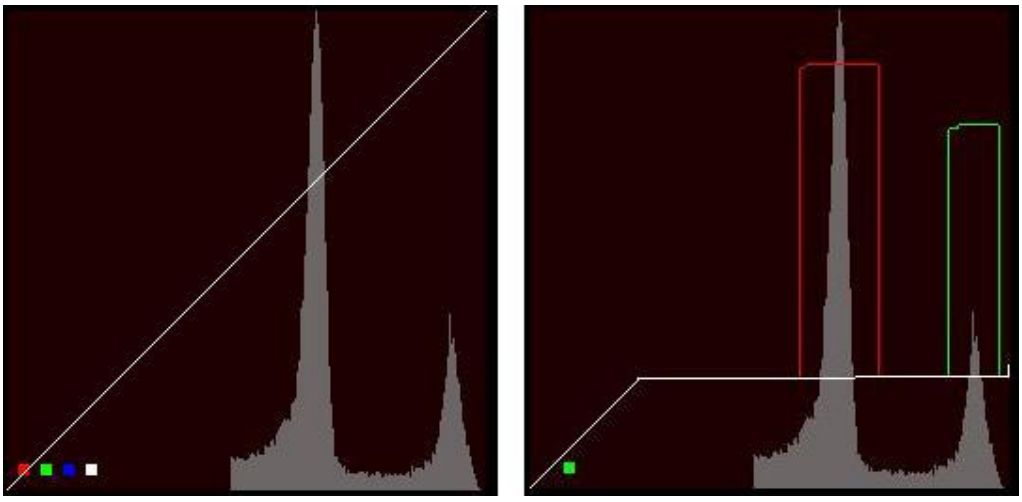


Figure 4.7 The interface showing the histogram and histogram mapping function of the tooth data. The mapping function can be changed by using mouse for red, green, blue channels separately. The left image shows default settings, the right image shows updated settings.

The histogram mapping function can be viewed and changed from the user interface as shown in Figure 4.7. It is possible to hide or display the interface in the figure in the left bottom part of the view port. The user can change the mapping function separately for red, green, blue and alpha channels by activating or deactivating the channels. Active channels are shown on the left bottom part of the interface as small boxes. In Figure 4.7, the right image shows a case in which only green channel is currently active and the dominant color is red for the first peak and green for the second. Figure 4.8 illustrates the image obtained by using the mapping in the right image of Figure 4.7. The first peak in the histogram is caused by the voxel values in the lower part and on the surface of the upper part, while the second peak is caused by the inner upper part of the tooth. Figure 4.8 reveals the consistency between the histogram mapping and the color of the tooth rendered when the fact that yellow is obtained by composition of red and green.

By setting the mapping function of all channels to the same function, a colorless image is obtained. An example of this method is to change only the transparency of

the tooth to view the inner structures of the tooth. By changing the transparency images similar to X-ray outputs can be achieved.

Figure 4.8 The tooth obtained by using the mapping function in the right image of previous figure.

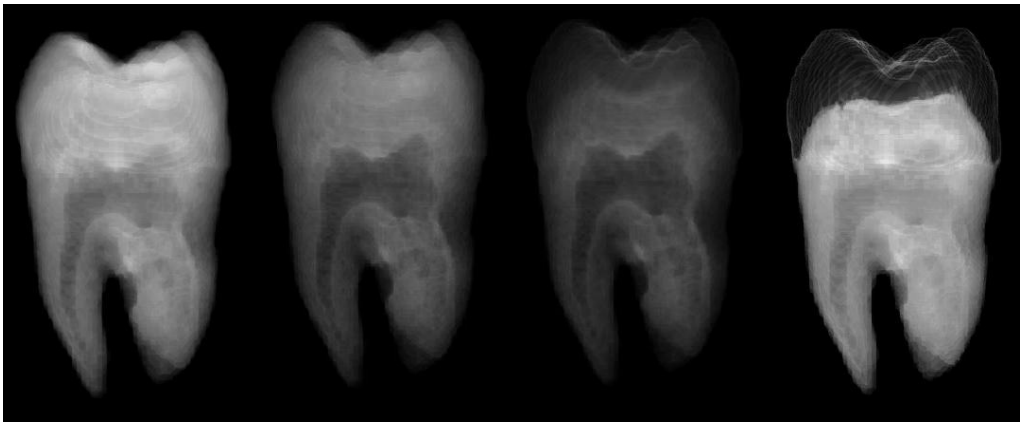


Figure 4.9 The tooth images obtained by using different transparency mapping functions.

4.3 PERFORMANCE COMPARISONS

As described throughout the chapter, different graphical rendering methods are implemented to render the deformable tooth. However, the entire jaw model is

rendered as a surface data. In order to compare the frame rates of different methods the deformable tooth is rendered alone on the screen and the refresh rates of the program is measured.

The different graphical rendering methods implemented to render the deformable tooth are: Surface rendering (marching cubes) on CPU, iso-surface rendering and volume rendering on GPU by using ray casting algorithms.

The frame rates are measured by using a computer with Intel Xeon 3.6 GHz CPU, 3.00 GB RAM and a NVIDIA Quadro FX 4500 graphics card. The operating system used is Windows XP and the size of the active rendering screen without the GLUI interface is 780x690. The refresh rates of the graphical rendering methods can be observed in Table 4-1.

Table 4.1 The refresh rates of different graphical rendering methods.

Rendering Method	**Frame Rates (Per sec.)**
Surface Rendering on CPU	156
Iso-surface rendering on GPU	28
Volume Rendering on GPU	24

As observed in Table 4-1 there is a big difference between the frame rates of the GPU based methods and the surface rendering method on CPU. The main reason behind this difference is the per pixel lighting calculations in GPU methods. The lighting calculation is evaluated for each pixel on the screen considering the gradient of the data on GPU based methods. Whereas, for the surface rendering on CPU the calculated lighting values for vertices of the triangles are interpolated to render the triangle surface. The tooth images obtained by the methods can be found in Figure 4.10.

Figure 4.10 The tooth images used to measure the frame rates for different graphical rendering methods. The left one is obtained by using surface rendering on CPU, the middle one represents iso-surface rendering on GPU and the right one represents the volumetric rendering on GPU.

CHAPTER 5

HAPTIC RENDERING AND DATA MODIFICATION

The integration of haptic devices to apply force and get feedback (force) to/from the user started a new era in virtual reality field. There has been a comprehensive study to achieve realistic forces by using haptic devices for more than a decade. However there are some hardware restrictions like maximum stiffness, maximum force that the haptic devices can provide. New haptic devices are being developed while a serious study is going on about the algorithms to evaluate realistic forces by current haptic devices in the field. In this chapter, we will discuss the methods of evaluating haptic forces (named as haptic rendering) in the literature, the method used in the thesis and data modification algorithm. The modification of the data and haptic rendering is closely related, because they share the same data and the effect of modification should be reflected to the haptic rendering in real time.

5.1 SURVEY OF LITERATURE

The haptic studies in the literature, some of which include the modification of an object in virtual environment, are presented below.

In the first study, a framework for training-oriented simulation of temporal bone surgery is developed in Stanford University. A hybrid data representation is used

that allows smooth surfaces to be maintained for graphic rendering while volumetric data is used for haptic feedback in the group's studies [8,9,28]. To evaluate the haptic forces the tool is discretized into a voxel grid and an occupancy map for the tool's voxel array is computed. At each interactive time step, each volume sample in the tool is checked for intersection with the bone volume. A sample point inside a bone voxel generates a unit-length contribution to the overall haptic force vector that tends to push this sample point toward the tool center, which with adequate stiffness is always outside the bone volume. Thus, overall penetration depth based on the number of immersed sample points is computed, and the overall force is generated as an oriented vector that is the sum of the contributions from individual sample points. This force's magnitude increases with the number of sample points immersed in the bone volume. The modification of the data is handled by decreasing the density values of the bone voxels which is occupied by the tool voxels. When bone voxels are updated, the hybrid data structure requires that the region around the removed bone be re-tessellated for graphical rendering. Consequently, bone voxels are queued up by haptic rendering thread as they are updated, and the graphic rendering thread re-tessellates the region around each voxel in this queue. That is, for each removed voxel, the software recognizes which of his neighbors have been revealed and creates triangles that contain the centers of these new voxels as vertices.

In another study [12], which concerns the probing and cutting operations in dental surgery, the forces are calculated by an empiric method. Both the tooth and tool data are represented as surface triangles. A force model composed of a frictional and resistive component is designed to implement experiments that identify the parameters of the force model. The force model considers different factors like the type and the rotational speed of the tool, the stiffness and the layer of the tooth, active contact area. The coefficients of these factors in the force model are set by measuring the forces the dentists experience in the setup prepared. The cutting operation occurs when the force evaluated exceed a threshold value. The triangle mesh is updated for graphical rendering. Vertex deformation method is used to update the tooth model as described in [11]. First the vertices on the tooth triangle

mesh that intersects with the tool is found, then all these vertices are extended along the vector N, which is the inverse direction of vector n(the normalized vector sum of all normals of triangles which intersects the tool), until the extending ray intersect with the tool sphere surface.

A group in Hamburg represents the data volumetrically and the tool as a cloud of sample points on a spherical structure in their studies [29, 30]. In haptic loop, each sample point on the tool surface is checked whether it collides with the data. For each point that immerses the data, a ray is traced through the center of the spherical tool until the data surface or the center of the tool is reached. All found vectors are added to find the force vector which must be applied to the haptic device.

Another algorithm for the haptic rendering introduced in the study [31] is a constraint based method which allows contacts to be determined between any point on the surface of the tool and any point on the surface of a virtual object in the scene. The strategy of the algorithm is to ensure that a tool, the visual tool, always stays on the surface of the objects while a second tool, the haptic interface tool, related to the haptic device is allowed to penetrate the objects. Initially the visual tool and the haptic interface tool will be located at the same position. Once a collision is determined between the haptic interface tool and an object in the scene the visual tool is constrained to the contact point on the surface of the object. This stage is determined by finding the pair of closest points on the surfaces of the two intersecting objects, using SOLID, a 3D collision detection library [32]. Then a spring damper model is used to model a force between the two representations of the tools. As force is evaluated by using the spring damper model in every haptic loop, the haptic interface tool is tracked continuously to render the visual tool appropriately constrained to the surface of the intersected object. The visual tool is moved to the closest point on the surface to the haptic interface tool. This stage is performed as long as the haptic interface tool remains in collision with the object. This constraint based method, named as “proxy based method” or “virtual tool method” or “god object method” [33], allows stable forces depending on the

penetration depth of the tool into the surface without visual artifacts of the tool penetrating the surface of the objects in the scene.

The idea of god object method is used together with the sampling point representation of tools in [34]. Dynamic objects are represented by a set of surface point samples, plus associated inward pointing surface normals. When a point of dynamic object interpenetrates a voxel, a depth of interpenetration is calculated as the distance of the point to the surface of the static data. The depth is used in the spring model to calculate the force of each point. The net force and torque acting on the dynamic object is obtained as the sum of all force/torque contributions from such point-voxel intersections.

5.2 HAPTIC RENDERING

God object method [33] which models a spring damper system between a virtual and a physical tool is used for haptic rendering. The algorithm was first implemented using the OpenHaptics HDAPI library [35], a low level haptic library. However, due to the instability a higher level haptic library, the OpenHaptics HLAPI [35] which provides more stable forces is used in the simulation.

The main idea of the implemented method [36] depends on rendering two separate tools: *the physical tool* and *the virtual tool*. The orientation and the position of *the physical tool* are set identical to the position and the orientation of the haptic device. However, *the virtual tool* is not allowed to immerse into the tooth data and a spring system is defined between *the virtual tool* and *the physical tool*. The user feels a contact force proportional to the distance between *the physical tool* and *the virtual tool* while *the virtual tool* is rendered on the surface of the tooth data as in Figure 5.1.

The position and the orientation of *the virtual tool* are set identical to *the physical tool*'s position and orientation if there is no collision between *the virtual tool* and the tooth. Collision is checked by an open source library called RAPID [37] when

the HDAPI is used. The working principle of RAPID library is described in Appendix C. In case of collision the first contact point of the tool with the tooth is labeled to be used to update *the virtual tool*'s position. The algorithm used to update *the virtual tool*'s position is illustrated in Figure 5.2.

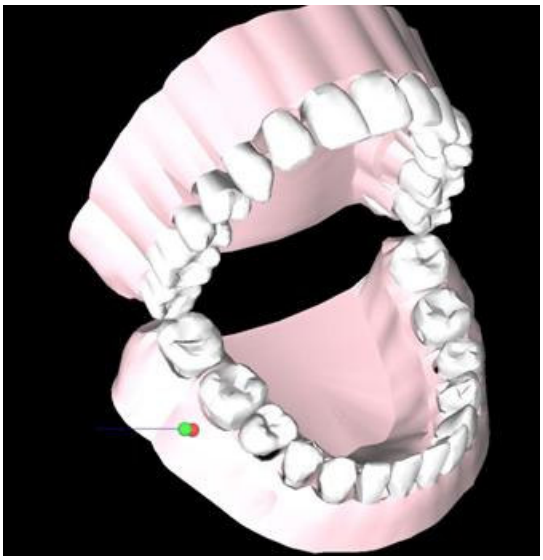


Figure 5.1 The Position of the virtual tool (green sphere) and the physical tool (red sphere) on the surface of the model.

The algorithm stores the previous collision state in the boolean variable "collisionFlag". In every haptic loop the collision flag is checked. If the tool is currently in collision with the tooth data, only *the virtual tool*'s position is updated such that *the virtual tool* will be on the surface point that is nearest to *the physical tool* . If "collisionFlag" is false, meaning that there is no collision in the previous loop, collision is checked between the tooth and *the virtual tool*. In case of no collision *the virtual tool*'s position is set to *the physical tool*'s position, which means the haptic device is in free space. If a collision is detected *the virtual tool* is prevented to immerse into the tooth surface and its position is set to the point that is nearest to *the physical tool*'s position on the tooth surface.

Updating the position of *the virtual tool* on the surface of the tooth uses the first contact point on the tool. The nearest point to the first contact point on *the physical*

tool is found on the tooth surface and *the virtual tool*'s position is set to this point. Figure 5.3 illustrates the algorithm used to update the position of *the virtual tool*.

```
if CollisionFlag is false
{
      if there is a collision with virtual
      position
      {
            Set CollisionFlag true
            Update position of the virtual tool
            on the surface of the tooth
      }
      else
      {
            Set virtual tool's position
            identical with the physical tool's
            position
      }
}
else
{
      Update position of the virtual tool on the
      surface of the tooth
}
```

Figure 5.2 The Pseudo code of the algorithm used to calculate the position of the virtual tool.

Before the collision occurs, *the physical tool* (Pt) is outside the object surface and itd position is identical *with the virtual tool* (Vt) (see Pt (t-3), Pt (t-2), and Pt (t-1)). When the Pt penetrates into object at time t, the Vt is constrained to stay on the surface. At time t+1, Pt moves to a new location (Pt (t+1)), and the new location of Vt is determined from the current Pt and its neighboring primitives based on the nearest distance criterion.

The algorithm described above is first implemented by using the Openhaptics HDAPI library [35]. However, the algorithm also exists in Openhaptics HLAPI library [35] which is a high level haptic library. It is observed that the forces achieved by HLAPI show more stabile characterization than the forces obtained from HDAPI. Discontinuities, vibrations and pulses are observed in the algorithm

implemented on HDAPI while the HLAPI provides smoother forces. For these reasons the HLAPI library is used for haptic rendering in the developed simulation.

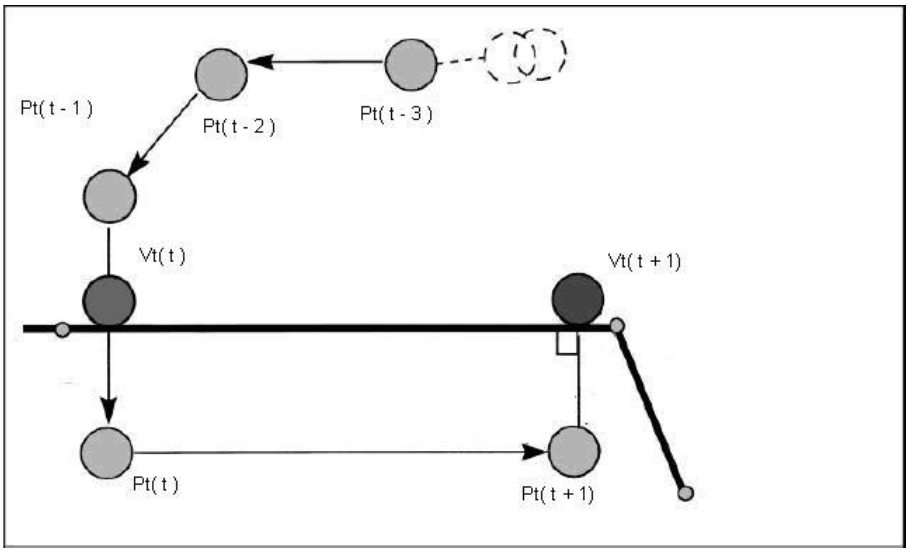


Figure 5.3 The illustration of the algorithm used to calculate the position of the virtual tool.

In HLAPI, it is possible to change some parameters that are used in the evaluation of the forces. As mentioned before the algorithm defines a spring model between *the physical* tool and *the virtual tool*. The stiffness and the damping coefficient of this spring model can be adjusted to a desired value. By changing the spring model's coefficients the biomechanical differences of different structures of the jaw model can be considered in the force calculations. The adjustment of the biomechanical properties of decay, dentine, enamel and palate is described in Chapter 2. Moreover, the static and frictional coefficients of the surfaces can be taken in to account. In case of collision, the structure that the haptic device is currently in contact is detected. Then the spring model's stiffness and damping coefficient, the static and dynamic friction coefficients are adjusted to the corresponding values of the structure being touched.

5.3 DATA MODIFICATION

After the diagnosis of the decay region in the tooth, our aim is to remove the decay region. In order to simulate the grinding operation of the decay, the density values of 3D tooth data are modified for the regions collided with the grinding tool.

The user can modify the tooth data if one of the following two conditions is satisfied.

- If the distance between *the physical tool* and *the virtual tool* exceeds some threshold value, which also means if *the physical tool* immerses into the surface of the tooth more than a specified deepness, the tooth data is modified. This method allows the user to modify the tooth data if he/she applies a force greater than a threshold value. The threshold values for modification are adjusted separately for different structures such as decay, dentine and enamel. Deepness threshold values for the immersion of *the physical tool* are used in the simulation. These threshold values are set as 2.0 N / corresponding stiffness coefficient for dentine and enamel structures, 0.2 N / corresponding stiffness coefficient by a collaborative work with the dentists from METU Health Center and Ankara University. The stiffness coefficients are the HLAPI stiffness variables without a unit. Their range is between 0.0 and 1.0, 1.0 represents the maximum stiffness that the haptic device can provide. By these threshold values, decay can be modified easily compared to the healthy structure of the tooth. The difference between the stiffness coefficients of dentine and enamel is sufficient to simulate the hardness difference between them.

- Another condition is added to the above to simulate the effect of the spherical grinding tool that rotates around itself. Because of the rotational velocity of the tool around itself, some grinding occurs even the tool is held constant on the surface of the tooth in real life. In the simulation if the user keeps the tool on the surface of the tooth constantly more than a specified

frame time, 1/3 of the deepness threshold value defined in the previous condition is applied for modification. By decreasing the deepness threshold it becomes easier to grind the surface layer of the tooth data with a rotating spherical tool. However in case the user tries to immerse into tooth the deepness threshold in the first condition is applied.

In case the modification algorithm is called, the density values of the tooth voxels collided with the voxels of the spherical tool are set to '0'. This operation is illustrated in Figure 5.4. After setting the corresponding voxels' values to '0', the gradient information of the neighborhood voxels is updated. The necessary updates for the graphical rendering depend on the rendering method used.

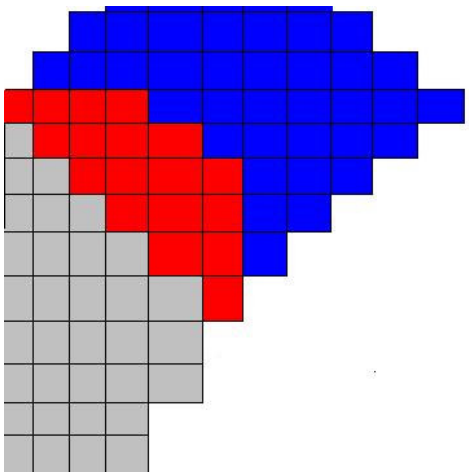


Figure 5.4 The modification operation. The gray voxels represent the tooth voxels, the blue ones represent the tool voxels. The red voxels show the tooth voxels collided with the voxels of the spherical tool.

If surface rendering on CPU is active, the modified voxels are added to a list called *Modified Voxel List*. After updating all the colliding voxels, the triangles of the voxels in the *Modified Voxel List* are deleted from the *Surface Display List*, the triangles which are rendered in every graphical loop. Then the marching cubes algorithm [20] is run for every voxel in the *Modified Voxel List* to evaluate the new triangular surface in the modified region. New triangles calculated by the algorithm are added to the *Surface Display List* and the *Modified Voxel List* is cleared.

If one of the graphical rendering methods on GPU is active, a bounding box which includes the modified region is specified. Then the values of the updated tooth data in this bounding box are sent to the graphical card to update the 3D texture used for graphical rendering.

5.4 VIBRATION

In addition to the haptic forces evaluated during the grinding operation of the tooth, a vibration effect is also integrated to the system to increase the level of realism. The user feels a vibration along the y axis with a frequency of 500 Hz and amplitude of 0.3 N during the grinding operation. It is observed that the vibration has a considerable effect to reach realistic haptic forces.

CHAPTER 6

STEREOVISION AND AUDIO EFFECTS

Each eye of the human beings takes a view of the same area from a slightly different angle. Each eye's views have plenty in common, but each eye picks up visual information the other does not. Each eye captures its own view and two separate images are sent to the brain for processing. These two images are united into a final image inside the brain. The brain matches the similarities between the two images and adds slight differences between the images to the final image. The combined image is more than the sum of two distinct images; it is a 3D stereo image. With stereovision we can see where the objects are with respect to our bodies more precisely, especially when they are moving toward or away from us in depth dimension.

The integration of the stereovision into virtual environments is not hard in today's technology. There are types of stereo glasses which are used to view the stereoscopic image rendered on the screen. Besides current graphic cards, which include built-in support for stereoscopic buffering, along with a standardized OpenGL interface support for the card's stereo features, make it easier to add stereovision to Windows-PC based applications.

The effects addressing to the 'hearing' sense of human beings is indispensable in computer games and other virtual environments. It is a fact that every computer

system has sound cards and speakers to play any sound. In development phase, it is not hard to use the audio features of the computer. Therefore the sound effects have been used in virtual environments throughout the history of virtual applications.

One of the aims of the thesis is to achieve a higher level of realism in the virtual environment by implementing stereovision and sound effects. In this chapter we will discuss the application of these effects in the simulation.

6.1 STEREOVISION

To adopt the stereovision into the simulation environment, stereoscopic viewing eyewear, CrystalEyes3 (Figure 6.1) is used. The product is composed of an eyewear part and an emitter. To achieve 3D images, the environment is rendered as stereo on the screen and the user views the screen by the eyewear. The eyewear shutters left and right eyes view with a frequency consistent with the refresh rate of the screen.

Figure 6.1 CrystalEyes3, stere3D eyewear and its emitter from StereoGraphics.

The developed simulation has the ability to render the graphical scene in stereo mode if the graphics card and the monitor support stereo vision. Graphics cards that support stereoscopic buffering are able to simultaneously maintain both left- and right-eye graphics buffers. This means that, in combination with standard double-buffering (in which graphics are drawn to a "back buffer" and then swapped to the "front buffer" when all drawing is completed), the board will typically maintain four buffers when in stereo display mode.

The “glDrawBuffer()” OpenGL function allows the developer to specify which buffer subsequent OpenGL drawings and renderings should be directed to. With normal non-stereo OpenGL double-buffering, the developer will typically draw to the "back buffer" (glDrawBuffer(GL_BACK)), and then "swap buffers" in order to put what had been drawn to the back buffer onto the "front buffer", which represents the visible display.

Once the stereo vision support of the hardware is checked the scene is rendered in stereo mode by using left and right buffers. The developer can specify the drawing buffer as GL_BACK_LEFT or GL_BACK_RIGHT, which will result in subsequent OpenGL drawings and renderings appearing only in one eye's drawing buffer. After drawing to both the left and right back-buffers, a single “swap buffer” function is called to put the back stereo buffers' contents to the front (visible) stereo buffers.

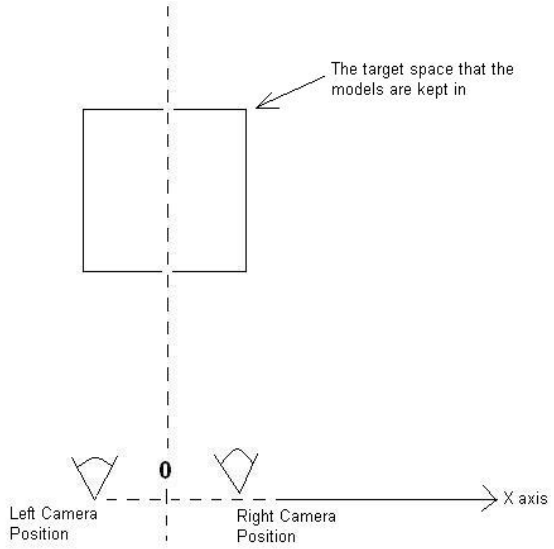


Figure 6.2 The left and right camera positions are adjusted along X axis. Then the frustum of viewpoints is adjusted so that for each eye’s view the models in the target space are rendered on the screen.

When drawing the left eye’s view to the back-left buffer, the camera position is adjusted so that the camera is moved towards left on the X axis by half the amount

of the opening between the two eyes. Then the perspective of the viewpoint is updated so that the left eye looks at the same target. To do this, the rectangular box, representing the limits of the space that will be rendered on the screen, specified by "glFrustum" function is moved toward right on the X axis by a specified amount. This process is illustrated in Figure 6.2. Similarly when the right eye's view is drawn to the back-right buffer, the camera is moved to the right on the X axis by half the amount of the opening between the two eyes and the frustum is adjusted.

6.2 AUDIO EFFECTS

In addition to the visual effects and force feedback audio effect capability is adapted to the simulation. The sound effects become active when a triggering event rises. There are 14 different wav files of actual screams of a person showing annoyance and pain. In the case when the user applies a force greater than a specified value or tries to drill the palate of the mouth, a triggering event for these sound effects rises. One of the sound files is chosen randomly and played. Moreover a grinding sound effect which becomes active during the grinding operation of the tooth is integrated to the system. This effect works only if the user is using the active grinding tool. The sound starts when the tool touches the tooth and stops when the tooth is untouched.

CHAPTER 7

IMPLEMENTATION

In this chapter we will discuss the libraries and the programming language used in the simulation developed for this thesis. The structure and the flowchart of the software will also be described. At the end of the chapter the user interface will be introduced and some screenshots from the simulation will be given.

7.1 THE LIBRARIES AND PROGRAMMING LANGUAGE

The libraries used in the thesis are: OpenHaptics for haptic rendering and OpenGL for 3D graphical rendering.

The OpenHaptics is the product of Sensable [38] company which also produces the Phantom haptic devices used in the simulation. The library supports both Windows and Linux environments. It has two components [39]: HDAPI (Haptic Device API) and HLAPI (Haptic Library API). HDAPI is a low level library which allows the developer to modify the hardware values of the haptic device. It is possible both to evaluate the force and moment values of the device and set the values of motors and encoders of the device. The HLAPI is a high level library working together with OpenGL graphics library. It is composed of three different threads: Client thread (~30 Hz), collision detection thread (~100 Hz) and haptic thread (~1000 Hz). The client thread accomplishes the graphical rendering and transfers geometrical

information to the collision detection thread. After the collision detection thread detects for the collisions in a local region, the force rendering operation is evaluated in the haptic thread. OpenHaptics library provides the ability to interfere the device in low levels and to develop haptic supported applications compatible with OpenGL.

For graphical rendering, OpenGL which is compatible and independent of the operating systems is used. By using OpenGL one can develop applications independent of the model of the graphical card and the architecture of the processor. It is also possible to reach several examples and documentation about OpenGL since it has become so common in the field.

The haptic library, OpenHaptics supports C++ as a programming language. Also because of the performance and optimization of C++, C++ is generally used in haptic simulations. Therefore it was decided to use Microsoft Visual Studio.NET C++ for the development of the application for the thesis.

7.2 THE STRUCTURE OF THE SOFTWARE

The main structures modeled in the virtual environment are the dental tool, the surface jaw model and the deformable tooth. For each of these structures a different class is implemented. Beside these classes the simulator has a graphical user interface module and a utility class. The modules of the dental simulator are illustrated in Figure 7.1. Some explanation for each class is given below:

The Graphical User Interface: This is the main module of the software and it is responsible for initiating the other modules and OpenGL settings. First the user interface is created at the beginning of the program and the settings for the OpenGL parameters are done. Then a separate object for *Jaw Model*, *Tooth* and the *Dental Tool* is created and initialized. All of the mouse and keyboard interactions by the user are handled by this module.

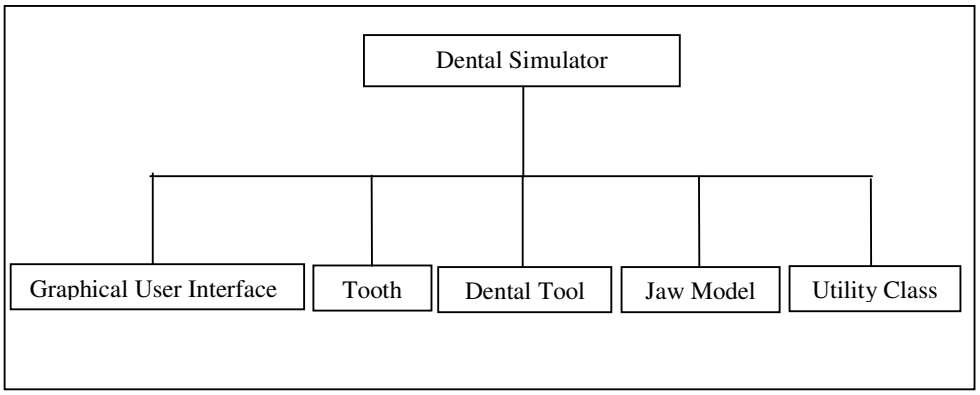


Figure 7.1 The modules of the dental simulator.

The Tooth: This is the class that represents the deformable tooth and stores the properties associated with it. The volumetric data, surface triangles or 3D texture, the biomechanical parameters, the place and the size of the decay etc... are kept in this class.

Dental Tool: The dental tool is represented by this class. The surface triangles for graphical rendering, the type (modifying or non-modifying) and the voxels of the modifying part of the tool are stored in the class.

Jaw Model: The entire jaw model (upper and lower palate and corresponding teeth) is modeled in this module.

Utility Class: This is the module for the general functions and constants used by all other modules. This includes the "glm" library [22] for the loading operations of the "obj" surface models. It also has a math component that includes all basic mathematical and matrix operations.

In order to achieve realistic reliable results using the haptic (phantom) device, the program has to be divided into two separate threads. This is because the manufacturer of the haptic device recommends the operation of the program loop responsible for the haptic force feedback at 1000 frames / sec. This frame speed is not possible in the OpenGL loop because of the complexity of the structures

rendered in OpenGL environment. Typically 30-60 frames / sec for OpenGL loop is sufficient for real time simulations. The flow chart of the program using the two threads is shown in Figure 7.2.

At the start of the application, the standard initializations for the GLUT [40] and GLUI (GLUT-based C++ user interface library which provides controls such as buttons, checkboxes, radio buttons, and spinners to OpenGL applications) libraries are done. After the jaw model is loaded by glm [22], the program makes the OpenGL settings and loading of the tooth data depending on the active graphical rendering method. The initialization functions for OpenGL are different for GPU and CPU methods. Also tooth data is loaded into different data types depending on the graphical rendering method. If the CPU based surface rendering is active, the surface list of the tooth data in 3D voxel array is created. Whereas, for GPU rendering methods, a 3D texture created from the voxel array is sent to the graphical card. The tooth is scaled and translated to be placed into the missing surface tooth model in the entire jaw model. Finally, the model for the dental tool is loaded and parameters for it are initialized before the display and haptic threads start working.

After the initializations of the structures, the display and haptic threads start to work. These threads work continuously until termination of the application. The frequency of the haptic thread has to be at least 1000Hz to render continuous and stabile forces for the haptic device. It is impossible to reach that frequency for the graphical thread because of the complex structures rendered on the screen. Also that frame rate is unnecessary since human eye can not distinguish the difference between the refresh rates greater than ~ 30 Hz.

In the beginning of the display thread, the deformable tooth is updated in case of modification. The modification of the 3D tooth array is a complex operation which needs a computational workload. Therefore it is not possible to implement modification in haptic thread which has to reach a frequency greater than 1000Hz to provide stabile and continuous forces. Collision detection should be done in display thread to decide whether the tooth will be modified or not. The collision detection

can be done by using the HLAPI's function that can work in client thread (the display thread of the simulation) without an inconsistency with the haptic thread. In case of a modification the tooth data which is used by both the display and haptic threads will be updated. Therefore this update operation has to be thread-safe in order to achieve force feedback consistent with the graphical rendering. By implementing the modification algorithm in the client thread of HLAPI which corresponds to the display thread in the simulation, the shared tooth data is updated without any inconsistency. After the modification of the tooth, the jaw model, the dental tool, and the deformable tooth are rendered graphically in the display thread. The haptic thread checks whether there is a collision between the dental tool and the jaw model or deformable tooth. The collision is checked in collision detection thread of HLAPI with a frequency of 100Hz. However the forces are calculated and transferred to the haptic device in haptic thread of HLAPI (1000Hz). The two main threads of the simulation continue working until the user exits the application.

7.3 THE USER INTERFACE

In the user interface the 3D environment including the jaw model and dental tool is rendered with a menu on the right side of the screen. The menu can be used to adjust the biomechanical properties of structures, change the dental tool, change rendering technique and activating or deactivating the haptic device. The keyboard shortcuts are also explained in the menu. In the title bar the graphical refresh rate can also be observed. Some screenshots of the user interface during can be viewed in the figures at the end of this chapter.

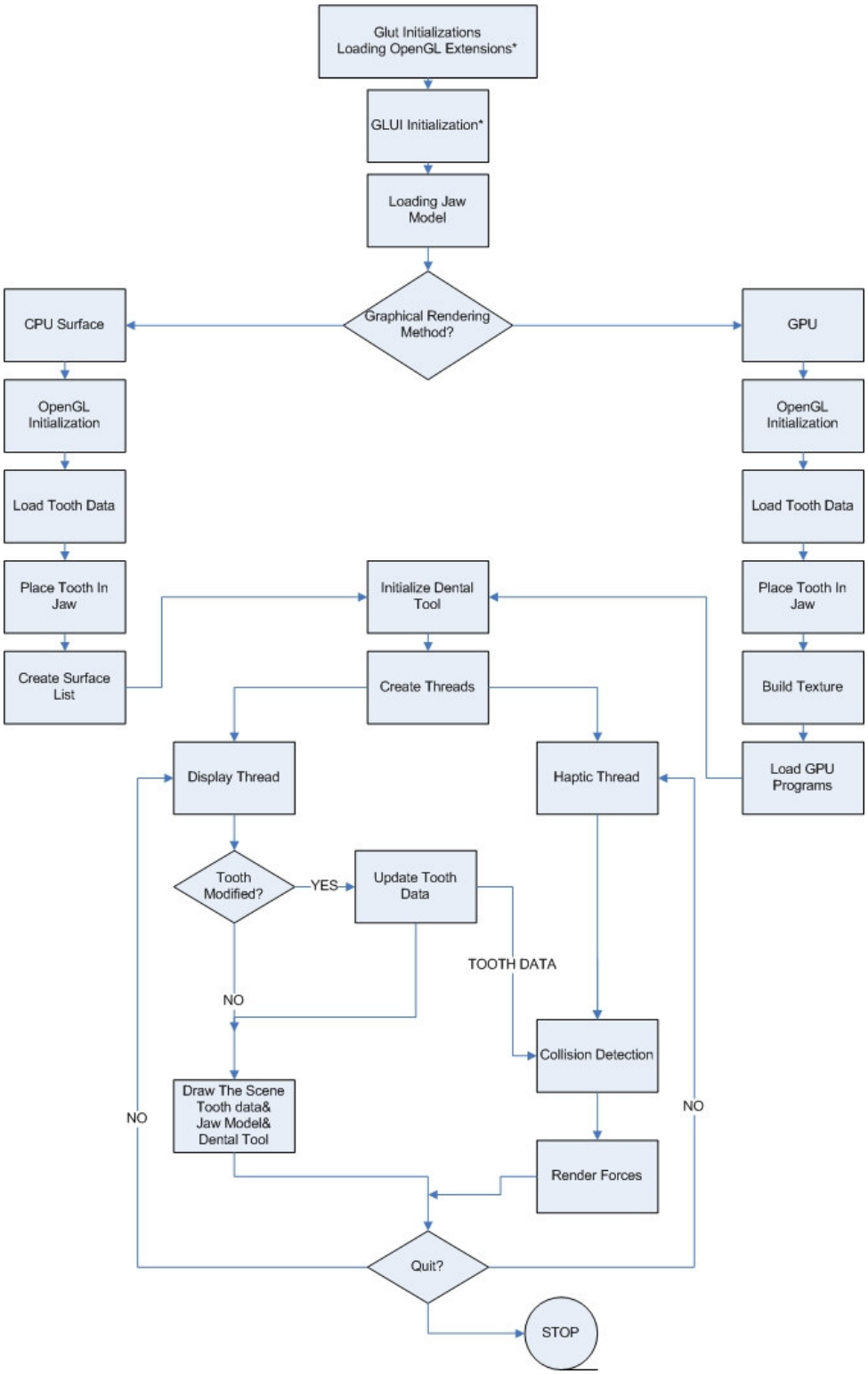


Figure 7.2 The flowchart of the dental simulator. The steps with '*' are the standard initializations of the libraries.

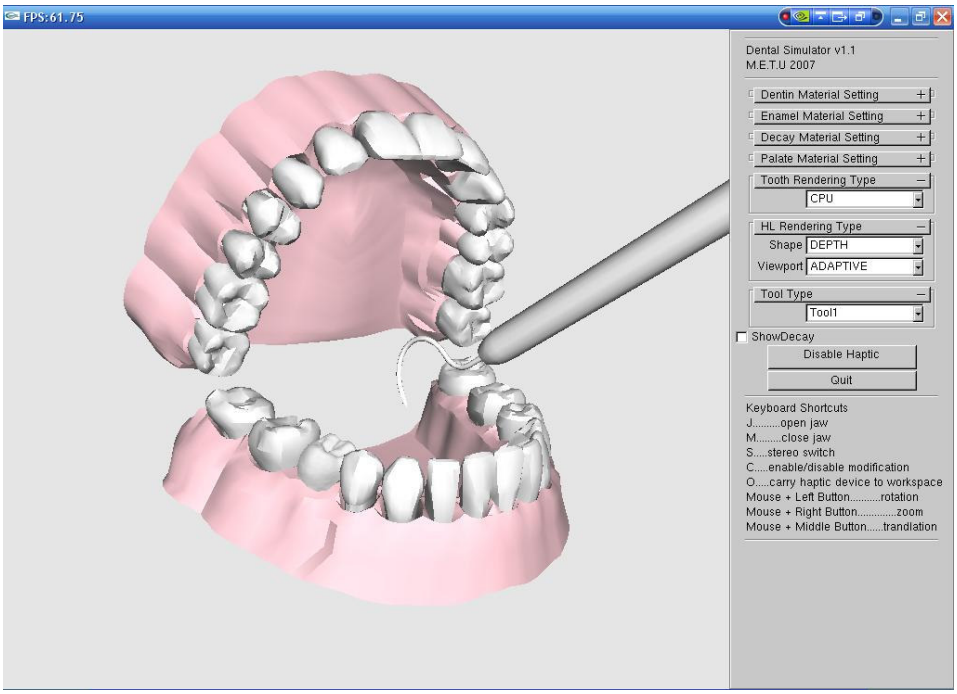


Figure 7.3 A Screenshot from the software.

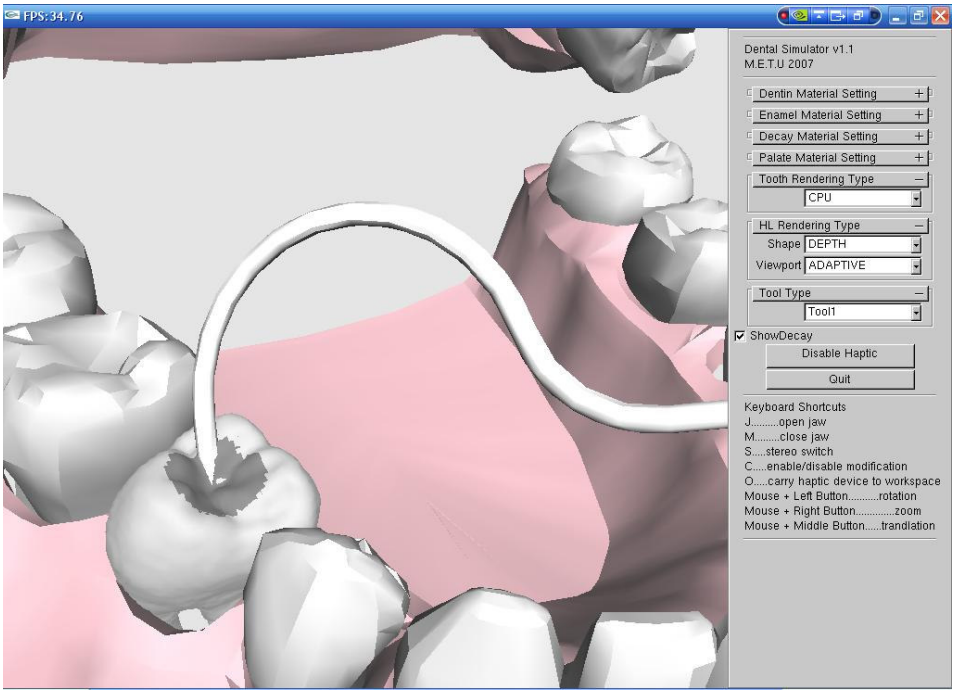


Figure 7.4 A Screenshot from during the diagnosis of decay.

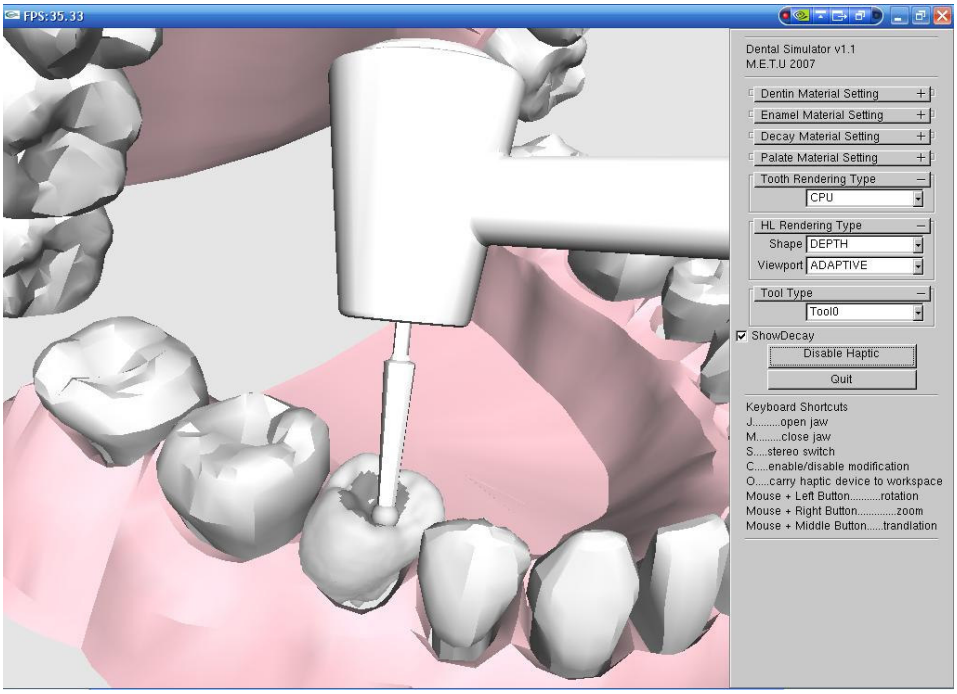


Figure 7.5 A Screenshot before grinding.

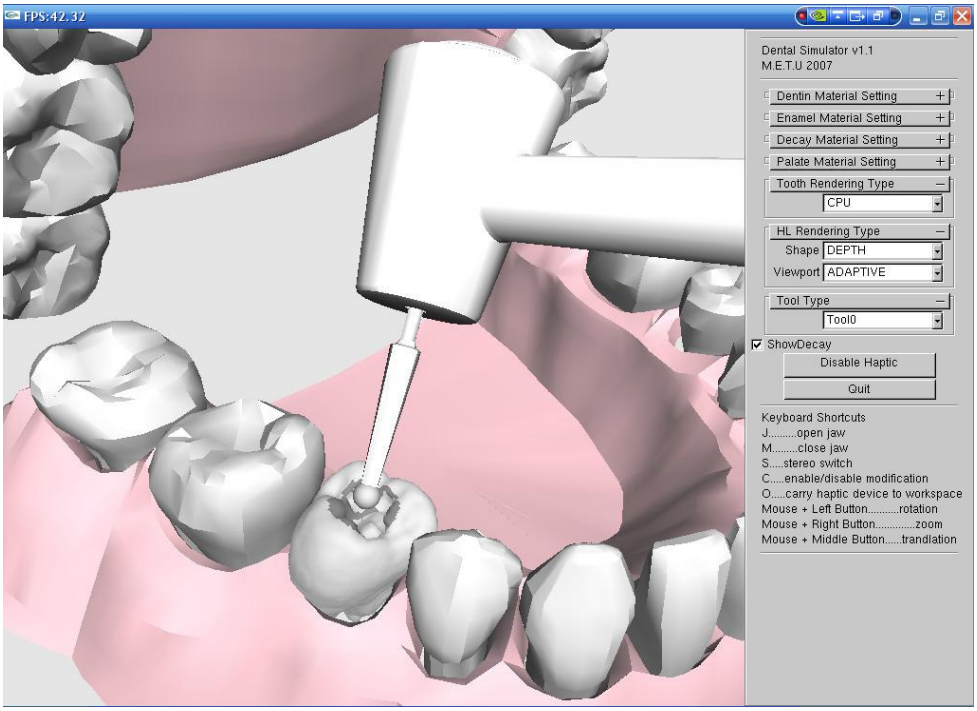


Figure 7.6 A Screenshot during grinding.

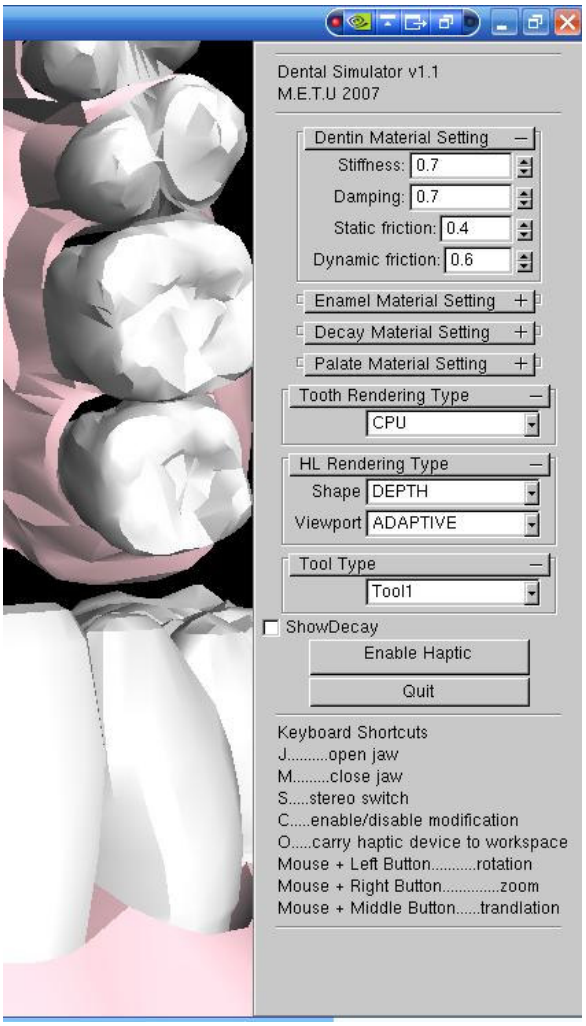


Figure 7.7 A Screenshot of the menu.

CHAPTER 8

CONCLUSION AND DISCUSSIONS

In the thesis, a virtual dental simulation environment with a haptic device is devised and developed. The environment consists of a screen, a computer, a haptic device, a stereovision eyewear and the software. By the integration of the haptic device the user can touch and modify the structures modeled in the environment. The stereovision capability provides more realistic visualization with depth information of the objects by creating 3D images.

The entire jaw is modeled and decay is placed in one of the teeth in the jaw. The user can examine the whole model with biomechanical differences between structures (palate, dentine, enamel, decay) by touching with haptic device. The decay can be located in different parts of the tooth to create different scenarios. The user is expected to detect the decay which has different biomechanical properties than healthy structures. A parametric system is developed such that the user can adjust the biomechanical properties of each structure by using the interface for experimental studies. After the user finds the decay, the next scenario is to carve the decay by using a different dental tool. Vibration during grinding and audio effects are integrated to the simulation to achieve more realistic feedback to the user.

After the system is developed, in order to asses the performance of the system a collaborative work with dentists from METU Health Center and Ankara University

and researchers from METU Industrial Design Department is done. An experiment procedure and questionnaire is prepared to take some measurements and some feedback from dentists. This experiment procedure is applied to 10 dentists from Ankara University.

In this chapter the method applied in performance tests will be described and the results will be discussed. At the end of the chapter, a general summary and possible future work will be discussed.

8.1 PERFORMANCE TESTS

In order to detect the missing features, measure the performance of the simulation, and decide on the future work it is decided that the system should be tested by dentists. The dentists' performance is measured in the simulation for improving the dental simulation environment in the future phases.

To prepare an evaluation procedure, the methods used for virtual environments are surveyed. After this survey, a study [41] which is generally used in the assessment of virtual applications in the field is found. A standardized questionnaire is applied to the users of the virtual environment in this study. However it is observed that this questionnaire is prepared considering virtual applications like computer games. Discussions on the method to apply for performance tests are done together with the dentists (from METU Health Center and Ankara University) and researchers from METU Industrial Design Department. As a result of these discussions, a questionnaire which includes statements about the simulation is prepared. The user is asked in which degree he aggress or disagrees to each statement. The user assigns a value to each statement between '1' and '5'. '1' represents the strong disagreement and '5' represents the strong agreement to the statement. The statements are prepared by using the knowledge obtained from [41]. The questionnaire is given to a dentist from Ankara University to understand whether the statements are clear for a dentist (a professional from medical field). The

questionnaire is corrected and finalized according to the advices and reactions of this dentist during the trial usage.

After the questionnaire is prepared and an experiment procedure is standardized, the system is setup in Ankara University Faculty of Dentistry and the experiment procedure is applied to 10 dentists. These 10 dentists are chosen from the ones who have interest on technology and are familiar with computer games. After the experiment procedure the dentists are given the questionnaire which can be viewed in Appendix D. The experiment procedure is described below.

8.1.1 EXPERIMENT PROCEDURE

During the Scenario 1 and 2, support about the use of the haptic device and simulation was given. However, during Scenario 3, 4, and 5 there was no intervention to the user.

Scenario1: The Introduction of Haptic Device:

- The placement of the user, the locations of screen and haptic device with respect to each other is standardized.
- The user is told that the system is in development phase.
- By showing the user the simulation including the jaw model, a documentation explaining the shortcuts and the usage of the software is given to the user.

o A period of time is allowed to the user to adjust a comfortable positioning. [Time: 0.5 Min.]

o The functions of the program described in the documentation are accomplished by the user. [Time: 5 Min.]

Scenario 2: The Introduction of the Decay:

- Two teeth, one of which has a decay, are rendered on the screen. The color difference between the decay and the healthy tooth structure is shown.

- A standard stiffness difference is adjusted between the decay and the healthy tooth structure.
  - A period of time is allowed to the user to recognize the tactile difference between the decay and healthy structures of the tooth. [Time: 5 Min.]
  - A period of time is allowed to the user to recognize the difference between grinding the decay and healthy structures of the tooth. [Time: 5 Min.]

Scenario 3: The Detection of the Stiffness Difference Level of Recognizing the Decay:

- Two teeth, one of which has a decay, are rendered on the screen without showing the color difference between the decay and the healthy tooth structure.
- The user is told that one of the teeth has a decay and the structure of the decay is same as the previous scenario.
- The stiffness difference between the decay and healthy structure is increased in steps until the user detects the decay region. While the stiffness of the healthy structure is kept constant as '1' unit, the stiffness of the decay is adjusted as '0.7', '0.4', '0.1' respectively.
- The level of the stiffness that the user detects the decay and the time for detecting the decay for that level is recorded.

Scenario4: The Diagnosis of the Decay by Using the Entire Jaw Model:

- The jaw model and the tooth which has a decay are rendered on the screen. The color difference between the decay and the healthy tooth structure is not shown.
- The stiffness of the decay is standardized for all of the users. The decay is located on one quadrant of the tooth surface randomly.
- The user is told that the tooth with a decay is in which region of the jaw without telling which tooth has the decay.
  - A period of time is allowed to the user to detect the decay and tell on which quadrant of the tooth surface the decay exists. [Time: 5 Min.]
- Whether the user detected the decay correctly or not is recorded. If the decay is detected correctly, the time for detecting the decay is also recorded.

Scenario5: The Grinding of the Decay by Using the Entire Jaw Model:

- The jaw model and the tooth which has a decay are rendered on the screen. The color difference between the decay and the healthy tooth structure is shown.

o A period of time is allowed to the user to clean the decay by grinding. [Time: 5 Min.]

- The time for cleaning operation is recorded. The remaining amount of decay, and the amount of healthy structure is saved at the end of the operation.

The listed objectives symbolized with '' represent the operations done by us, and the objectives symbolized with 'o' represent the ones that the user does. The aim of Scenario1 is to adapt the user into the environment and to teach the user the features of the software developed. By this process, the users become familiar to the haptic devices and get used to the virtual environment. Scenario 2 proposes to demonstrate how the difference between the decay and healthy tooth structure is modeled in the simulation. Also the user gains experience of grinding the decay by haptic device in this scenario. In the 3rd scenario without showing the color difference between the decay and the healthy tooth, an experiment about the accuracy of the stiffness difference between decay and healthy tooth is done by using only tactile feedback. The aim is to understand above how much stiffness difference the decay is detected on the tooth. In Scenario 4, the user tries to find the tooth with decay in the jaw model, and the quadrant of the tooth surface the decay lies on. The color difference is not shown and the time for detecting the decay is recorded in this scenario. In the last scenario, the user removes the decay and the time of the operation, the amount of the remaining decay and the amount if drilled healthy structure is recorded. After the experiment procedure the users fill the questionnaire in Appendix D.

8.1.2 THE RESULTS OF THE EXPERIMENT

In addition to the questionnaires, some assessments are done during the performances of 10 dentists as explained in the experiment procedure. These assessments can be observed in Table 8.1.

Table 8.1 The measurements during experiment procedure.

	Scenario3		Scenario4		Scenario5		
User No	**The stiffness of decay detected correctly**	**Time to detect decay (min)**	**The quadrant on which decay lies**	**Time to detect decay (min)**	**Time to clear decay (min)**	**The amount of cleared decay (%)**	**The rate of healthy tooth drilled to initial decay amount (%)**
1	0.4	1:35	4	5:00X	3:16	98.74	75.84
2	0.4	5:00	3	5:00X	5:00	97.75	96.89
3	0.7	1:12	2	1:20	5:00	72.92	47.88
4	0.1	0:40	1	3:56X	3:38	96.03	91.22
5	0.4	1:46	3	5:00	5:00	97.54	99.06
6	0.7	1:09	4	1:40	3:18	95.52	44.94
7	0.7	0:57	2	4:22	4:12	99.70	125.0
8	0.7X	1:38	3	1:05	4:30	97.40	67.89
9	0.4X	1:40	4	3:15	4:10	97.49	76.07
10	0.7	0:53	3	3:20	4:20	98.84	88.14

The ‘X’ signs in the result columns of Scenario2 and Scenario 3 show that the decay was not detected correctly.

In Scenario3, the user is shown two teeth one of which has decay on its surface. The 8^{th} and 9^{th} users thought that the healthy tooth had decay on it, while the other users could detect the decay correctly. The stiffness of the healthy tooth is set to ‘1’ and kept constant while the stiffness of the decay is set to ‘0.7’, and decreased to ‘0.4’and ‘0.7’ in steps until the decay is detected by the user. 4 of the users detect the decay correctly when the stiffness value of decay is ‘0.7’, 3 of them at ‘0.4’ and 1 of the users at ‘0.1’.

In the 4^{th} scenario, the decay is located on one of the quadrants of the tooth surface. When the tooth is viewed from above, ‘1’ represents the upper-left, ‘2’ represents the upper-right, ‘3’ represents the lower-left, and the ‘4’ represents the lower-right quadrant as illustrated in Figure 8.1. The number of voxels in the first quadrant is 1538, 1374 in the second, 1383 in the third and 1275 in the fourth quadrant of the tooth surface. 7 of the users detected the decay in the correct quadrant and the

average time spent to detect the decay correctly is calculated as 2:00 from Table 8.1.

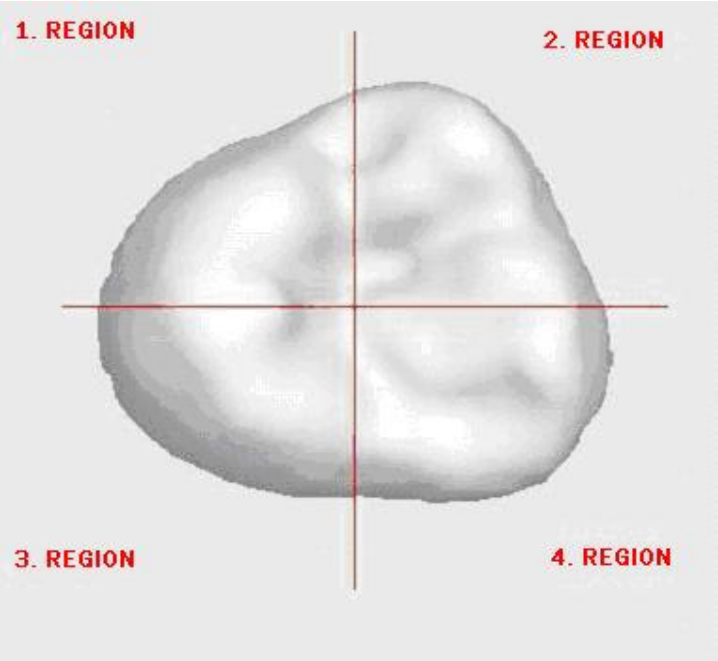


Figure 8.1 The quadrants of the tooth surface.

In the 5th scenario, the users try to remove the decayed region without harming the healthy structure. The decay region has different number of voxels (volume) in each quadrant of the tooth surface because of the curved structure of the surface. Therefore the amounts of the remaining decay and the healthy tooth structure drilled are calculated as a percentage by dividing them with the initial amount of the decay. It is observed that % 95.19 of the initial decay is cleared as average, while an amount of healthy tooth that is equal to %81.29 of initial decay is also is drilled. It can be calculated that the average time to remove the decay is 4:14.

8.1.3 THE RESULTS OF THE QUESTIONNAIRE

After the performances of the users, they filled a questionnaire (Appendix D) which measures the usability, clarity, effectiveness, supportiveness of the system and the satisfaction of the users from the system. The questionnaire consists of statements

about the simulation. The users assign a value from ‘1’ to ‘5’ to these statements. ‘1’ shows strongly disagreement and ‘5’ shows strongly agreement to the statements. To evaluate the results of the questionnaire the average of these values for each statement is calculated. However there are negative statements about the software together with positive statements in the questionnaire. Therefore the symmetric of the negative statements’ values according to the average value ‘3’ are used in general evaluation for negative statements. For instance, an assigned value ‘1’ to the statement: “This system has no advantage for my hand-on experience.” shows that the user does not agree to the statement. This shows that the user thinks that system has advantages for hand-on experience. The assigned value ‘1’ is considered as a ‘5’ for general performance of the system. The questionnaire examines the satisfaction of the user in six different categories below.

The Usability: The usability can be defined as how much a product can satisfy the expectations of the user. The usability shows how much useful the product is for the user after his/her experience.

The Clarity: Clarity can be defined as the user’s capability to understand the interface and the components of the interface (controls, buttons, warnings etc.), to guess the next step during the interaction with the environment. A “clear” product can be used without a learning phase according to this definition.

The Effectiveness: One can describe effectiveness as accomplishing the aimed task by using minimum amount of resources. The resources spent while using the product can be considered as time, number of progresses and cognitive or physical effort. Therefore in an effective virtual environment the user should be able to accomplish the tasks rapidly without unnecessary steps.

The Help/Support: This can be defined as the ability of the user to solve the problems, ambiguities and to continue the tasks without any help from outside. In the evaluation of this feature, we try to understand whether the user interface of the simulation has a supportive structure or not.

The Satisfaction: In general, this is the measurement of the satisfaction level of the user from the simulation. In this aspect, the facts like how much the user is attracted from the simulation, whether the simulation bothers the user or not are evaluated.

The performance of the simulation is calculated over 5 for every question and average values for each category are found. The average values of each question in the questionnaire can be found in Appendix E. The average values for each category can be viewed in Figure 8.2 below.

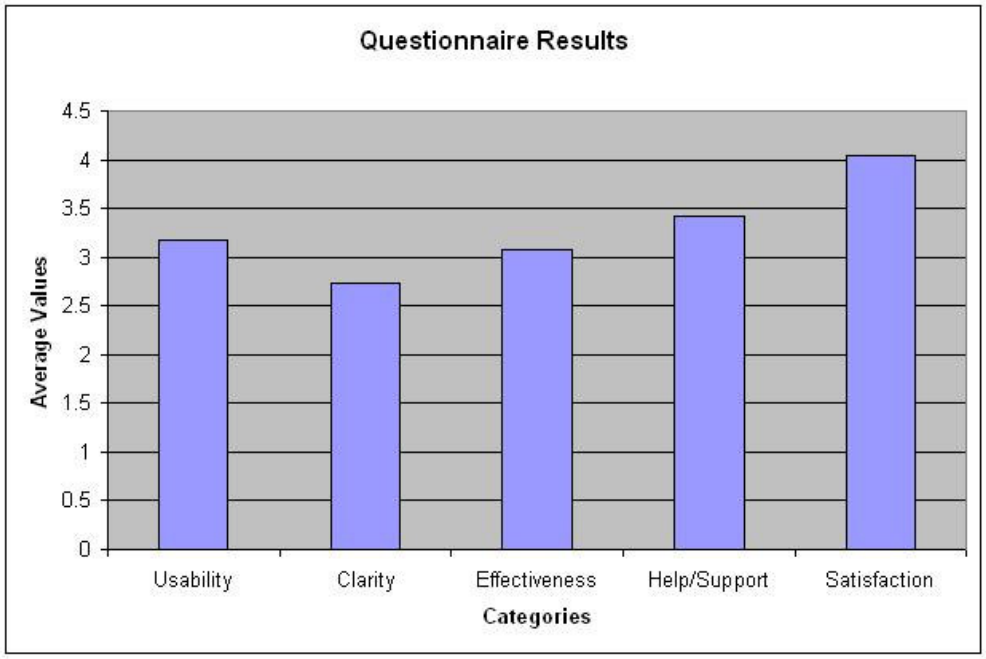


Figure 8.2 The average values of the results of the questionnaire.

When the results are evaluated, it can be observed that the system may need an improvement in "Clarity" category. When the results for clarity are examined in a detailed way, one can say that the only criterion below the average is "the similarity of the feelings of simulation to reality". The users mentioned that they could feel the structure of the tooth, see the details and find the reactions apprehensible. Therefore it can be said that the subjective low value for this statement is not actually a negative opinion. The reason behind it is the fact that the virtual environment includes different conditions from the real experiences of the dentists.

In the “Effectiveness” category, a performance below the average is observed about the practical and easy usage of the system. When it is considered that all of the users see and use the haptic device first time for the experiment, one can say that this low performance is expected. If the performance of this category remains the same after some more experiments then it can be said that there is a problem in the system.

The satisfaction of the users from the system is above average and this affects the motivation of the users. Similarly, the facts that they do not need support and find the system usable show that the system reached its aims.

8.2 DISCUSSION AND FUTURE WORK

The haptic technology has created a new era in the virtual world by providing the capability of touching and feeling in the virtual environment. The effect of integration of haptic technology into medical field triggered evolutionary approaches to vocational education technologies. Haptic applications are threatening the classical approaches especially in the fields where hands on training is not applicable or ethical like dentistry.

The apprenticeship, learning from someone else, has become the most popular method for years in dental education. Also plastic tooth models placed into a jaw model are used for hands on experience before facing a real patient. The simulations were not popular in dental training field because of the absence of the tactile feedback. After the integration of haptic technology to simulations, a trend of using simulations has started in the field. The dental simulations supported by haptic feedback have gained superiority over traditional methods. In aspects of reality, performance assessment, demonstration and cost-effectiveness the simulations provide several advantages. The automatic performance assessment, repeatable usage of the data, the ability to create scenarios and the simultaneous usage of the same scenario are some of the advantages over traditional methods.

In this thesis, a dental virtual environment which allows basic dental applications is developed. The entire jaw is modeled and a tooth with decay is placed in the jaw model. A parametric model is developed to adjust the biomechanical properties of the layers of the tooth, decay and palate separately. Two scenarios are created in the simulation: The detection of the decay and removing the decay region by using different dental tools. After the system is developed, it is tested by dentists to take feedback and asses the performance of the system. These tests showed that the dentists have a great motivation of using the system although it may need to be improved in aspects of clarity.

The dental simulation environment is developed considering the integration of different modules and functions. As a future work,

- A 6 DOF(Degree of Freedom) haptic rendering,
- The filling operation of the decay region with amalgam after the decay is removed,
- The integration of a second haptic device as an irrigation or suction tool,
- The modeling of soft tissues like cheek and tongue

can be implemented to achieve a more functional simulation environment.

REFERENCES

[1] Ranta, J.F. and Aviles, W.A. (1999). "*The Virtual Reality Dental Training System—Simulating Dental Procedures for the Purpose of Training Dental Students Using Haptics*", Fourth PHANTOM Users Group Workshop.

[2] Phantom Head Dental, "Ivorine Models", Retrieved August 28, 2007 from http://www.phantomhead.com/columbia/brochure/ivorinemodels.htm

[3] DenX, "Advanced Dental Systems," , Retrieved August 28, 2007 from http://www.denx.com/dentsim_system_desc.html

[4] European Patent Office, "Methods and apparatus for simulating dental procedures and for training dental students", Retrieved August 28, 2007 from http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=US2002119432&F=0

[5] Thomas, G., Chen, J., Johnson, L., Stanford, C. (2002). "Iowa Dental Surgical Simulator" Retrieved from Iowa University, Information Technology Services website, Retrieved August 28, 2007 from http://at.its.uiowa.edu/atac/awards/2001/dental-surg-sim-update.pdf

[6] "Use of Haptics In Dentistry", University of Illinois at Chicago, Retrieved August 28, 2007 from http://www.uic.edu/classes/dadm/dadm396/ADSreserch/Intro.htm

[7] "Simulation of Bone Surgery," Stanford Biorobotics Lab., Retrieved August 28, 2007 from
http://jks-folks.stanford.edu/bonesim/

[8] Morris D, Sewell C, Barbagli F, Blevins N, Girod S, Salisbury K. (2006) "*Visuohaptic Simulation of Bone Surgery for Training and Evaluation.*", IEEE Transactions on Computer Graphics and Applications, *pp. 48-57.*

[9] Morris D, Girod S, Barbagli F, Salisbury K. (2005) "*An Interactive Simulation Environment for Craniofacial Surgical Procedures.*" Proceedings of MMVR (Medicine Meets Virtual Reality).

[10] Sewell C, Morris D, Blevins N, Agrawal S, Dutta S, Barbagli F, Salisbury K. (2007) "*Validating Metrics for a Mastoidectomy Simulator.*" Proceedings of MMVR (Medicine Meets Virtual Reality).

[11] D.X. Wang Y. Zhang Y. Wang P. Lu and Y. Wang. (2003) "*Development of Dental Training System with Haptic Display,*" Proc. IEEE Int'l Workshop Robot and Human Interactive Comm. (RO-MAN 2003) *pp. 159-16.*

[12] Liu G., Zhang Y., and Wang D. (2005) "*Cutting Force Model of Dental Training System.*" In Proceeding of IEEE IROS 2005, *925—929.*

[13] Philip F.Ostwald, Jairo Munoz.. (1997) *Manufacturing Processes and Systems*. pp *270-330.*

[14] myDr., "Structure of a normal tooth" , Retrieved August 28, 2007 from http://www.mydr.com.au/default.asp?article=3728

[15] Angker, L., V. Swain, M., N. Kilpatrick. (2003) "*Micro-Mechanical characterization of the properties of primary tooth dentine,*" Journal of Dentistry *31,261-267.*

[16] Bell TJ, Bendelli A, Field JS, et al. (1992) "*The determination of surface plastic and elastic properties by ultra-microindentation.*", Metrologia *28:463–9*.

[17] E. Mahoney, A. Holt, M. Swain, N. Kilpatrick (2000) "*The hardness and modulus of elasticity of primary molar teeth: an ultra-micro-indentation study*," Journal of Dentistry *28, 589-594*.

[18] Giannini, M., Jose´ Soares C., M. Carvalho, R. (2004) "*Ultimate tensile strength of tooth structures,*" Dental Materials *20,322-329*.

[19] Balijepalli, A., Kesavadas, T. (2003) "*A Haptic Based Virtual Grinding Tool,*" IEEE Virtual Reality Symposium 2003, March 22-26, Los Angeles, CA.

[20] William E. Lorensen, Harvey E. Cline. (1987) "*Marching Cubes: A high resolution 3D surface construction algorithm.*" In: Computer Graphics, Vol. *21, Nr. 4.*

[21] SenseGraphics AB, Retrieved August 28, 2007 from www.sensegraphics.se

[22] Robin, N. " Open GL Tutors", (2000). Retrieved August 28, 2007 from www.xmission.com/~nate/opengl.html.

[23] Kılıç V., Koçak U., Konukseven E. İ., Mumcuoğlu E. (2006). "*GPU Supported Haptic Device Integrated Dental Simulation Environment,*" EUROHAPTICS 2006 Conference, *p.135-140.*

[24] Stegmaier S., Strengert M., Klein T., and Ertl T.(2005). "*A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-based Raycasting*", Volume Graphics pp. *187-195*.

[25] Klein T., Strengert M., Stegmaier S., and Ertl T. (2005). "*Exploiting Frameto-Frame Coherence for Accelerating High-Quality Volume Raycasting on Graphics Hardware*", Procceedings of IEEE Visualization '05 , *pages 223-230*., IEEE.

[26] T. Phong, B. (1973) "*Illumination of Computer-Generated Images*" Department of Computer Science, University of Utah, *UTEC-CSs-73-129*.

[27] K. Engel, M. Kraus, and T. Ertl. (2001) "*High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading*", In Eurographics / SIGGRAPH Workshop on Graphics Hardware '01.

[28] Morris, D., Sewell, C., Blevins C., Barbagli F., Salisbury, K. (2004) "*A Collaborative Virtual Environment for the Simulation of Temporal Bone Surgery,*" Proceedings of the International Conference on Medical Imaging Computing and Computer Assisted Intervention (MICCAI), St. Malo, France, Springer Lecture Notes in Comptuer Science, *Vol. II, pp. 319-327*.

[29] Petersik A., Pflesser B., Tiede U., Höhne K., Leuwer R. (2002) "*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. Orlando, FL, *66-72*

[30] Petersik, A., Pflesser B., Tiede U., Höhne K., Leuwer. R. (2003) "*Realistic Haptic Interaction in Volume Sculpting for Surgery Simulation,*" In Nicholas Ayache, Hervé Delingette (eds.): Surgery Simulation and Soft Tissue Modeling, Proc. IS4TM 2003, Lecture Notes in Computer Science 2673, Springer-Verlag, Berlin, *194-202*.

[31] Laycock, S.D. and Day, A.M. (2003) "*The Haptic Rendering of Polygonal Models involving Deformable Tools*", EUROHAPTICS Dublin, pp. *176-192*.

[32] M.C. Lin and S. Gottschalk. (1998) "*Collision Detection between geometric models: a survey,*" In the Proceedings of IMA Conference on Mathematics of Surfaces.

[33] Zilles, C. B., Salisbury, J. K. (1995) "*A constraint-based god-object method for haptic display,*" Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. IEEE/RSJ International Conference, *On page(s): 146-151 vol.3.*

[34] McNeely, W. A., Puterbaugh, K. D., Troy, J. J. (1999) "*Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling*", Proc. ACM SIGGRAPH, 401-408.

[35] Sensable Technologies, Retrieved August 28, 2007 from http://www.sensable.com/documents/documents/OpenHaptics_datasheet_hi.pdf

[36] Ho, C., Basdogan, C., Srinivasan, M.A., (1999) "*An Efficient Haptic Rendering Technique for Displaying 3D Polyhedral Objects and Their Surface Details in Virtual Environments*", October'99 *Vol. 8, No. 5, pp. 477-491*, Presence: Teleoperators and Virtual Environments, MIT Press.

[37] Gottschalk, S., M. Lin and D. Monocha, "*RAPID -- Robust and Accurate Polygon Interference Detection*", Retrieved August 28, 2007 from http://www.cs.unc.edu/~geom/OBB/OBBT.html

[38] Sensable Technologies, Retrieved August 28, 2007 from www.sensable.com.

[39] Itkowitz, B., Handley, J., Zhu, W., SensAble Technologies, Inc®., USA, "The OpenHaptics™ Toolkit: A Library for Adding 3D Touch™ Navigation and Haptics to Graphics Applications", Retrieved August 28, 2007 from www.sensable.com.

[40] Kilgard, M.J., Silicon Graphics, Inc. "The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3".Retrieved August 28, 2007 from www.opengl.org.

[41] Witmer, B. G. and M. J. Singer (1998). "*Measuring presence in virtual environments: A presence questionnaire*." Presence: Teleoperators and Virtual Environments, 7(3): 225-240.

APPENDICES

APPENDIX A HAPTIC DEVICES

Specifications for the PHANTOM® Omni™ haptic device

The SensAble Technologies PHANTOM® product line of haptic devices makes it possible for users to touch and manipulate virtual objects. Different models in the PHANTOM product line meet the varying needs of both research and commercial customers. The PHANTOM Premium models are high-precision instruments and within the PHANTOM product line provide the largest workspaces and highest forces, and some offer 6DOF capabilities. The PHANTOM Desktop and PHANTOM Omni models offer affordable desktop solutions. Of the two devices, the PHANTOM Desktop device delivers higher fidelity, stronger forces, and lower friction, while the PHANTOM Omni device is the most cost-effective haptic device available today. Below are the specifications for the PHANTOM Omni haptic device.

Model	The PHANTOM Omni Device
Force feedback workspace	~6.4 W x 4.8 H x 2.8 D in ~ 160 W x 120 H x 70 D mm
Footprint Physical area the base of device occupies on the desk	6 5/8 W x 8 D in ~168 W x 203 D mm
Weight (device only)	3 lb 15 oz
Range of motion	Hand movement pivoting at wrist
Nominal position resolution	~ 450 dpi ~ 0.055 mm
Backdrive friction	<1 oz (0.26 N)
Maximum exertable force at nominal (orthogonal arms) position	0.75 lbf. (3.3 N)
Continuous exertable force (24 hrs.)	~ 0.2 lbf. (0.88 N)
Stiffness	X axis ~ 7.3 lb/in (1.26 N/mm) Y axis ~ 13.4 lb/in (2.31 N/mm) Z axis ~ 5.9 lb/in (1.02 N/mm)
Inertia (apparent mass at tip)	~0.101 lbm. (45 g)
Force feedback	x, y, z
Position sensing [Stylus gimbal]	x, y, z (digital encoders) [Pitch, roll, yaw (± 5% linearity potentiometers)]
Interface	IEEE-1394 FireWire® port
Supported platforms	Intel-based PCs
GHOST® SDK compatibility	No
3D Touch™ SDK compatibility	Yes
Applications	Selected Types of Haptic Research and The FreeForm® Concept™ system

SensAble technologies

SPECIFICATIONS COMPARISON FOR THE PHANTOM® PREMIUM 1.5/6DOF & 1.5 HIGH FORCE/6DOF HAPTIC DEVICES

	PHANTOM Premium 1.5/6DOF		PHANTOM Premium 1.5 High Force/6DOF	
Workspace	Translational	15 W x 10.5 H x 7.5 D inches 381 W x 267 H x 191 D mm	Translational	15 W x 10.5 H x 7.5 D inches 381 W x 267 H x 191 D mm
	Rotational		Rotational	
	Yaw	297 degrees / 5.18 radians	Yaw	297 degrees / 5.18 radians
	Pitch	260 degrees / 4.54 radians	Pitch	260 degrees / 4.54 radians
	Roll	335 degrees / 5.85 radians	Roll	335 degrees / 5.85 radians
Footprint	13 W x 10 D inches / 330 W x 254 D mm		13 W x 10 D inches / 330 W x 254 D mm	
Range of motion	Lower arm movement pivoting at elbow		Lower arm movement pivoting at elbow	
Nominal position resolution	Translational	860 dpi / 0.03mm	Translational	3784 dpi / 0.007 mm
	Rotational		Rotational	
	Yaw & Pitch	0.0023 degrees 0.00004 radians	Yaw & Pitch	0.0023 degrees 0.00004 radians
	Roll	0.0080 degrees 0.00014 radians	Roll	0.0080 degrees 0.00014 radians
Backdrive friction (x, y, z)	0.15 oz / 0.04 N		0.75 oz / 0.2 N	
Maximum exertable force and torque at nominal position (orthogonal arms)	Translational	1.9 lbf / 8.5 N	Translational	8.4lbf / 37.5N
	Rotational		Rotational	
	Yaw & Pitch	73 oz-in / 515 mNm	Yaw & Pitch	73 oz-in / 515 mNm
	Roll	24 oz-in / 170 mNm	Roll	24 oz-in / 170 mNm
Continuous exertable force and torque at nominal position (orthogonal arms)	Translational	0.3 lbf / 1.4 N	Translational	1.4 lbf / 6.2N
	Rotational		Rotational	
	Yaw & Pitch	27 oz-in / 188 mNm	Yaw & Pitch	27 oz-in / 188 mNm
	Roll	7 oz-in / 48 mNm	Roll	7 oz-in / 48 mNm
Stiffness	20 lbf in^{-1} 3.5 N mm^{-1}		20 lbf in^{-1} 3.5 N mm^{-1}	
Inertia (apparent mass at tip)	< 0.30 lbm < 136 g		< 0.46 lbm < 210 g	
Weight (device only)	~ 20 lb ~ 9 Kg		~ 20 lb ~ 9 Kg	
Force feedback	x, y, z, Tx, Ty, Tz		x, y, z, Tx, Ty, Tz	
Position sensing	x, y, z, roll, pitch, yaw		x, y, z, roll, pitch, yaw	
Interface	Parallel port		Parallel port	
Supported Platforms	Intel-based PCs		Intel-based PCs	
GHOST® SDK Compatibility	Yes		Upon special request	
OpenHaptics™ Toolkit Compatibility	Yes		Yes	

APPENDIX B STIFFNES & ELASTICITY OF MODULUS

Modulus of elasticity, E, can be calculated by dividing the tensile stress by the tensile strain:

$$E \equiv \frac{tensile\ stress}{tensile\ strain} = \frac{\sigma}{\varepsilon} = \frac{F/A_0}{\Delta L/L_0} = \frac{FL_0}{A_0 \Delta L} \qquad \text{(Equation 2)}$$

Where,

E is the modulus of elasticity (Young's modulus) measured in pascals;
F is the force applied to the object;
A0 is the original cross-sectional area through which the force is applied;
ΔL is the amount by which the length of the object changes;
L0 is the original length of the object.

Force exerted by stretched or compressed material

The modulus of elasticity of a material can be used to calculate the force it exerts under a specific strain.

$$F = \frac{EA_0 \Delta L}{L_0} \qquad \text{(Equation 3)}$$

Where,

F is the force exerted by the material when compressed or stretched by ΔL. From this formula can be derived Hooke's law, which describes the stiffness of an ideal spring:

$$F = \left(\frac{EA_0}{L_0} \right) \Delta L = kx \qquad \text{(Equation 4)}$$

Where,

$k = \frac{EA_0}{L_0}$ and $x = \Delta L$

APPENDIX C RAPID LIBRARY

This section explains the collision detection library, RAPID. Collision detection in this context stands for intersection detection which is to find out if two geometric entities intersect or not. The geometry of the colliding objects is the primary factor in choosing a collision detection algorithm. Bounding volume hierarchies (BVHs) collision detection algorithm is used in this study when HDAPI is used for its simplicity rapidness for performing collision detection on complex models.

A bounding volume hierarchy is a tree of bounding volumes, such as boxes. The computational goal of using BVHs is to minimize the time spent in determining if two objects do not intersect. A BVHs tree provides a multi-scale representation of the object. The root of the tree corresponds to an approximation of the object by a single sphere or box. The spheres or boxes corresponding to the middle levels of the tree represent smaller pieces of the object, thus providing a somewhat better approximation to the object than the root. The leaf nodes of the tree represent the actual geometry of the object.

A BVH based collision query proceeds by recursively testing bounding volumes of the model for overlap. For each such test, if they do overlap, then the children of the bounding volumes are tested pair wise for overlap. If they do not overlap, then that recursion branch terminates. If two leaf nodes are found to overlap, then the polygons they enclose are tested pair wise, and the results of these tests are added to the query output. This sequence of tests can itself be organized into a tree, which is shown pictorially in Figure C.1 below,

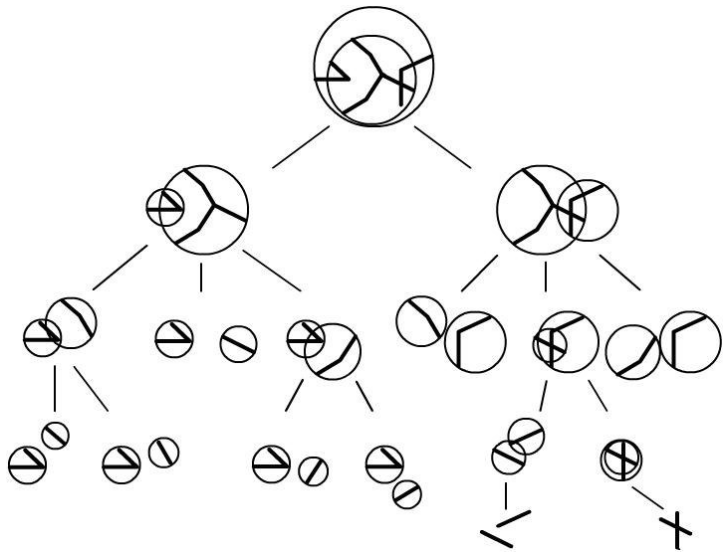


Figure C.1 Pictorial representation of the Bounding Volume Test Tree.

One of the important features of the RAPID Library beside its collision detection enquiries is its ability to find the closest distance between two objects. The UML model of the library main class is shown below.

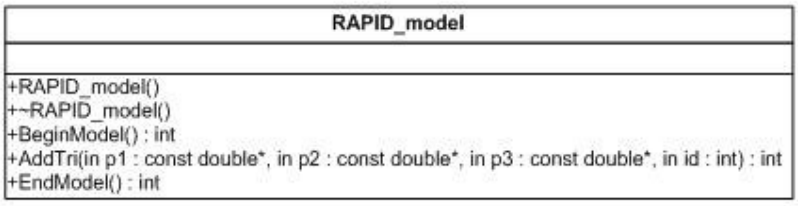


Figure C.2 UML Model of the RAPID_model class.

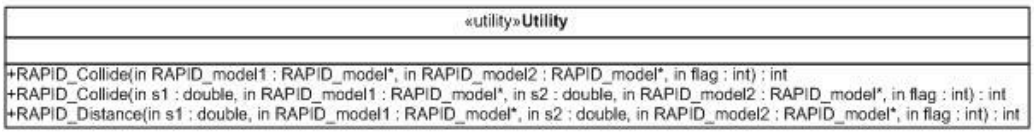


Figure C.3 UML Model of the RAPID Utility class.

APPENDIX D QUESTIONNAIRE

The Usability:

1. This system is very useful to improve my hand-on experience.
2. This system has no advantage for gaining hand-on experience.
3. I can use this system to practice in many ways.
4. I prefer improving my hand-on experience without using this system.
5. This system does not satisfy my expectations about improving my hand-on experience.

The Clarity:

1. I can feel the structure of the tooth while using the system.
2. The reactions of the system are apprehensible.
3. I can see the details of the tooth while using the system.
4. The feedback I feel in the system is close to my real experiences.
5. The feedback I feel in the system is not clear.

The Effectiveness:

1. I can accomplish what I want rapidly in this system.
2. I can not reach the tooth I choose easily.
3. I felt that I have the control in the system.
4. I could move inside the mouth easily in the system.
5. I do not need to perform unnecessary operations to accomplish a task.

The Help/Support:

1. System guides me for efficient usage.
2. System could not help me as much as I expected.
3. I often felt ambiguity about how to continue.
4. I may need anyone's help while using this system.
5. I can solve the problems I face in the system by myself.

The Satisfaction:

1. I find using the system interesting.
2. I was bored when using the system.
3. I want to learn more about the system.
4. I felt that I can manage the diagnosis of the decay and removing the decay region when I used the system.
5. It was fun to work with the system.

APPENDIX E QUESTIONNAIRE RESULTS

The Usability:

1. This system is very useful to improve my hand-on experience……...…2.8
2. This system has no advantage for gaining hand-on experience……...3.7
3. I can use this system to practice in many ways……3.6
4. I prefer improving my hand-on experience without using this system……3.2
5. This system does not satisfy my expectations about improving my hand-on experience……2.6

The Clarity:

1. I can feel the structure of the tooth while using the system……..3.3
2. The reactions of the system are apprehensible……..2.8
3. I can see the details of the tooth while using the system..……...…3.0
4. The feedback I feel in the system is close to my real experiences……...1.9
5. The feedback I feel in the system is not clear…...……..2.7

The Effectiveness:

1. I can accomplish what I want rapidly in this system……...……2.2
2. I can not reach the tooth I choose easily……....….3.0
3. I felt that I have the control in the system……....3.1
4. I could move in side the mouth easily in the system……...……3.7
5. I do not need to perform unnecessary operations to accomplish a task…...3.4

The Help/Support:

1. System guides me for efficient usage……..…...…2.9
2. System could not help me as much as I expected…….....3.2
3. I often felt ambiguity about how to continue……...3.9
4. I may need anyone's help while using this system……...……....3.6
5. I can solve the problems I face in the system by myself…….….3.5

The Satisfaction:

1. I find using the system interesting……...…….....3.9
2. I was bored when using the system…….…….....…4.3
3. I want to learn more about the system…….....…4.5
4. I felt that I can manage the diagnosis of the decay and removing the decay region when I used the system……...……..3.1
5. It was fun to work with the system…….….4.4

APPENDIX F USER GUIDE OF THE SOFTWARE

Menu Controls:

- **Activating the Haptic Device:** The device can be activated by using the "ENABLE HAPTICS" button.
- **Changing the Dental tool:** From the "Tool Types" panel in the menu, "Tool0" can be selected to clear the tooth and "Tool1" can be selected to diagnose the decay.

Changing the Viewpoint:

- By dragging the mouse while left button is pressed the jaw model can be rotated to change the view.
- By dragging the mouse while the right button is pressed, the jaw model can be zoomed in or out.
- By dragging the mouse while the middle button is pressed, the jaw model can be translated in the screen.

Keyboard Shortcuts:

All the keyboard shortcuts work only if the mouse is positioned in the active screen.

- **J, M:** The opening angle of the mouth can be changed by using these two buttons. The 'J' button is used to open and the 'M' button is used to close the mouth.

- **O:** In case of losing the position of the haptic device in the environment, haptic device can be brought to the workspace by pressing the ‘O’ button once. To release the haptic device after it is brought to the workspace ‘O’ should be pressed once more.
- **C:** To turn on and off the grinding tool, ‘C’ can be used.