

Human Head Stiffness Rendering

by

Minggao Wei

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© Minggao Wei, Ottawa, Canada, 2015

Abstract

The technology of haptics rendering has greatly enriched development in Multimedia applications, such as teleoperation, gaming, medical and etc., because it makes the virtual object touchable by the human operator(s) in real world. Human head stiffness rendering is significant in haptic interactive applications as it defines the degree of reality in physical interaction of a human avatar created in virtual environment. In a similar research, the haptic rendering approach has two main types: 1) Haptic Information Integration and 2) Deformation Simulation. However, the complexity in anatomic and geometric structure of a human head makes the rendering procedure challenging because of the issues of accuracy and efficiency. In this work, we propose a hybrid method to render the appropriate stiffness property onto a 3D head polygon mesh of an individual user by firstly studying human head's sophisticated deformation behaviour and then rendering such behaviour as the resultant stiffness property on the polygon mesh. The stiffness property is estimated from a semantically registered and shape-adapted skull template mesh as a reference and modeled from soft tissue's deformation behaviour in a nonlinear Finite Element Method (FEM) framework. To render the stiffness property, our method consists of different procedures, including 3D facial landmark detection, models semantic registration using Iterative Closest Point (ICP) technique, adaptive shape modification processed with a modified Weighted Free-Form Deformation (FFD) and FEM Simulation. After the stiffness property is rendered on a head polygon mesh, we perform a user study by inviting participants to experience the haptic feedback rendered from our results. According to the participants' feedback, the head polygon mesh's stiffness property is properly rendered as it satisfies their expectation.

Keywords. Multimedia haptics, stiffness property rendering, models semantic registration, adaptive shape modification, nonlinear finite element modeling.

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Professor Abdulmotaleb El Saddik, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

Secondly, my special and sincere thanks would go to Dr. Haiwei Dong, for his introduction to this research topic and suggestion. I have an invaluable lesson from his erudition and preciseness in academic.

I am also grateful to all colleague researchers in MCRLab for their advices and assistance in my project and thesis. My last thanks would go to my family, my girlfriend Kalin Mo, and all my friends for their loving support in me through all these years.

Table of Contents

List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Motivation	1
1.1.1 Haptics Technologies	1
1.1.2 Human Head Stiffness Rendering	5
1.2 Objective and Contribution	7
1.3 Thesis Organization	8
Nomenclature	1
2 Related Work	9
2.1 Facial Landmark Detection on a 3D Point Cloud	9
2.2 Models Registration	10
2.3 Shape Modification	11
2.4 Stiffness Rendering	11
2.4.1 Particle system	12
2.4.2 Continuum models	13
2.5 Haptic rendering	16
3 Background Information of Continuum Model and Finite Element Method	18
3.1 Continuum Model	18
3.1.1 Strain	19
3.1.2 Stress	20
3.1.3 Constitutive Law	20

3.2	Finite Element Method	21
3.2.1	Discretization	21
3.2.2	Assembly	23
3.3	Nonlinearity of the Finite Element Method	24
4	Methodology	26
4.1	System Architecture	26
4.2	Facial Landmark Detection and Models Registration	27
4.2.1	Facial Landmark Detection on a 3D Point Cloud	27
4.2.2	Models Registration	30
4.3	Adaptive Skull Model Approximation	31
4.3.1	Hierarchical ROI	32
4.3.2	Weighted Free-Form Deformation	33
4.4	Stiffness Rendering	38
4.4.1	Physical Volume Model	38
4.4.2	FEM Simulation	39
4.4.3	Head Polygon Mesh Stiffness Property Integration	40
5	Results	41
5.1	Experiment Setup	41
5.2	Landmark Detection and Models Registration Results	41
5.3	Skull Shape Adaption Results	42
5.4	Stiffness Rendering Results	44
6	User Study of Rendered Stiffness	47
6.1	Configuration	47
6.1.1	Hardware	47
6.1.2	Software	48
6.2	Deployment	48
6.3	Overall System Architecture	49
6.4	Component Illustration	50
6.4.1	Collision Detection	50
6.4.2	Coefficient Selection	51

6.4.3	Force Rendering	52
6.4.4	Haptic Update	52
6.4.5	Graphic Update	53
6.5	User Study	53
6.5.1	Participants	54
6.5.2	Procedures	55
6.5.3	Analysis	55
7	Conclusion and Future Work	62
7.1	Conclusion	62
7.2	Future Work	63
	APPENDICES	64
A	Human Head Mesh Stiffness Survey	65
A.1	Overview	65
A.2	Participant Background Information	66
A.3	Tutorial	66
A.4	Experiments	67
A.4.1	Physical-Interaction-Only Experiment	67
A.4.2	Physical-and-Visual Experiment	67
A.4.3	Rationality Evaluation	68
A.4.4	Comments and Suggestions	68
	References	69

List of Tables

4.1	Surface type classification table.	30
4.2	Material properties of flesh and bone type elements.	38
6.1	Hardware Configuration.	48
6.2	Software Configuration.	49
6.3	Ratings of region stiffness of overall and participant groups.	56
6.4	Rationality ratings of all participant groups.	56

List of Figures

1.1	Bidirectional flow of information in haptics.	1
1.2	Samples of haptic interfaces or devices.	2
1.3	A human operator is performing teleoperation with haptic devices. ⁷	3
1.4	Haptic feedback interfaces or devices for gaming.	4
1.5	A surgeon student is practicing operation on a surgical training system. ¹¹ .	4
1.6	Blocks of haptic rendering[1].	5
1.7	Structures of a human head in anatomy. ¹²	6
2.1	A sample 3D facial landmark detection. ¹³	10
2.2	A sample of 3D models registration. ¹⁴	10
2.3	Deforming a rigid object by control point manipulation. ¹⁵	11
2.4	Mass-spring system structure. ¹⁶	12
2.5	Liver modeled in a mass-spring system for haptic rendering; (a) is the pre-operative model and (b) is the intraoperative one. ¹⁷	13
2.6	Volume mesh of a continuum model; (a) is the surface of a volume mesh, and (b) is the cut plane of the volume mesh ¹⁸	14
2.7	Volume mesh element types; (a) is a tetrahedral element; (b) is a pyramid element; (c) is a prism element, and (d) is a hexahedral element.	14
2.8	Haptic interaction types: (a) is the point-type interaction ¹⁹ and (b) is the ray-type ²⁰	16
3.1	Relationships between the physical status in continuum model	19
3.2	FEM framework pipeline	22
4.1	Overview of system architecture.	26
4.2	Illustration of facial landmark detection and models registration procedures.	28
4.3	Example of a quadric surface with least square fitting generated from a 50-neighbour sample around nose tip on a 3D human head point cloud dataset. Points represent the input 50-neighbour.	29

4.4	Aesthetic proportions of corresponding features on the skull model.	31
4.5	The pipeline of the semantic ICP registration approach.	32
4.6	Adaptive approximation pipeline.	33
4.7	ROI segmentations and their hierarchy.	33
4.8	The STU local coordinates of the skull mesh. Its S is aligned with the X -axis, T with the Y , and U with the Z	34
4.9	The flesh and bone layer of our volume model.	39
4.10	Cut view of the classified elements of a tetrahedral mesh. The boundary between <i>bone</i> and <i>flesh</i> sets is presented with a solid line. Elements inside belong to the <i>bone</i> set while those outside are classified as the <i>flesh</i> set.	39
5.1	Detected facial landmarks on a 3D point cloud. The position of features are marked with a sphere. (a)-(d) are the detection result of user A from different views, and (e)-(h) are result of user B from different views. (a) and (e) are shown in isometric view. (b) and (f) are shown in frontal view. (c) and (g) are shown in profile view. (d) and (h) are shown in top view.	42
5.2	Results of models registration. (a) shows the facial features from detection results. (b) is the result of generated corresponding features on a skull point cloud. (c) displays the registered head and skull point cloud.	43
5.3	The local deformation result of the Jaw region obtained from the modified weighted Free-Form Deformation.	43
5.4	The fully adapted skull mesh as all ROIs are deformed. (a) is the original template skull mesh. (b)-(g) are the shape approximation results of two different users displayed in multiple views. (b)-(d) show the result of user A. (e)-(g) show the result of user B.	44
5.6	The simulation result of the features Cheek and Cranium are depicted with different markers respectively. The solid line is the distance between Cranium feature point and skull mesh, while the dash line shows the distance of Cheek feature.	45
5.5	Deformation behaviour of each selected feature modeled from FEM simulation: the values on horizontal axis is the applied external force (N), and on the vertical axis is displacement (mm). (a) illustrates the features on a polygon mesh. (b)-(h) are modeled behaviours of features (Cranium, Forehead, Eyebrow, Cheekbone, Cheek, Chin, and Jaw)	46
6.1	Hardware models of 3D scanner and haptic device. (a) is the 3D head scanner ²¹ . (b) is the haptic device ²²	47
6.2	The deployment of the Human Head Haptic Rendering Application.	49
6.3	Overall system architecture of the Human Head Stiffness Rendering System.	50

6.4	Intersection computation in a detected collision event.	51
6.5	The haptic tool of a Novint Falcon device and its position represented in Cartesian coordinates.	53
6.6	The Novint Falcon and other apparatus as configured in the user study. It shows a participant experiencing the rendered stiffness in a Physical-and-Visual Experiment. The screen is turned off temporally while performing a Physical-Interaction-Only Experiment to make sure no graphic information is provided in that experiment.	54
6.7	Average ratings of overall and for each participant group for each selected region.	57
6.8	Average ratings of regions from the Physical-Interaction-Only Experiment and Physical-and-Visual Experiment.	58
6.9	The distributions of participants' selection of tough and soft features. (a) is the distribution of the tough features selection. (b) is the distribution of the soft features selection.	60
6.10	Rationality ratings of overall and for each participant group. (a) is the distribution of rationality ratings of overall participants. (b) is the distribution of the experienced group. (c) is the distribution of the novice group.	61

Chapter 1

Introduction

1.1 Motivation

1.1.1 Haptics Technologies

Haptics, originating from the Greek for *to touch*, was initially introduced in the field of psychophysics in the early 20th Century as a way to label the subfield of studying human touch perception and manipulation [2, 1]. In the 1970s and 1980s, haptics was brought to robotics in order to build robots with self-ruling ability. Though the early exploration was frustrating, it inspired multiple mutual research topics, including *Human Haptics*, *Machine Haptics*, *Computer Haptics* and *Multimedia Haptics*. Srinivasan *et al.* [3] discussed human and machine haptics in their work on Human haptics aims to study human's perception and manipulation through touch, including tactile and kinesthetic; machine haptics refers to building interfaces and devices to facilitate the communication between human and computers. Computer haptics facilitates the physical interaction of virtual environments and objects through management of simulation algorithms and computations of proper interaction force. Multimedia haptics integrates and coordinates the presentation of haptic interface data and diverse sensory displays such as audio, video and text. As illustrated by El Saddik *et al.*, in terms of haptics technologies, the sense of touch is a unique modality given its bidirectional flow of information between human operator and virtual world [4] (see Figure 1.1).

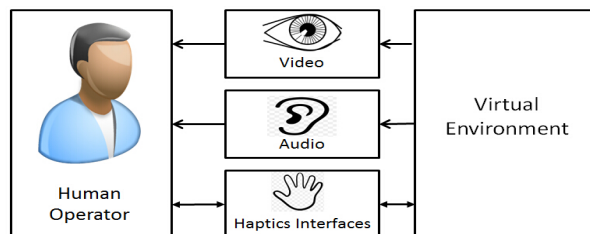
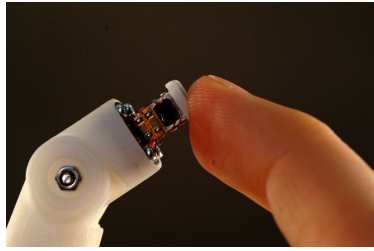


Figure 1.1: Bidirectional flow of information in haptics.



(a) Tactile haptics devices.¹



(b) Force feedback devices.²



(c) Ground-based haptic device.³



(d) Body-based haptic device.⁴



(e) Impedance haptic device.⁵



(f) Touchless haptic interface.⁶

Figure 1.2: Samples of haptic interfaces or devices.

Due to the focus on biomedical and psychophysical matters, human haptics will not be discussed in our research. As stated by Vincent Hayward *et al.* [5] and Salisbury *et al.* [1], haptics interfaces, or devices, can be classified by their displays (tactile or force feedback), grounding locations (ground-based or body-based), intrinsic mechanical behaviour (impedance or admittance), and amount of Degree of Freedom (DOF) of motion or force presented on devices or interfaces. With new technology, the modern haptic interface has the ability to produce touchless haptic feedback in a different manner, such as ultrasound, air vortices, or magnetic fields. Cross-sectional samples of the above haptic interfaces or devices are shown in Figure 1.2.

The breakthrough of computer haptics occurred in the early 1990s with the rise of several technologies that enabled displays realized by haptics interfaces or devices. Defined in the works of Srinivasan [6] and El Saddik [2], computer haptics focuses on the algorithms and software in simulating and rendering the touch of virtual object(s). Haptics rendering plays a critical role in computer haptics as it detects geometry contact information and

¹<http://www.wired.com/2013/02/haptics>

²<http://www.giantbomb.com/novint-falcon-support/3015-5149/>

³<http://www.senztech.cc/showpros.aspx?proid=35/>

⁴<http://www.it.bton.ac.uk/staff/lp22/CS133/haptics.html>

⁵<http://www.vrlab.ctw.utwente.nl/eq/HapticMaster.html>

⁶<http://www.bristol.ac.uk/news/2014/december/haptic-shapes-using-ultrasound.html>



Figure 1.3: A human operator is performing teleoperation with haptic devices.⁷

computes the proper interaction force between devices and a virtual environment [7]. Visual rendering, the other component in computer haptics, synchronizes real-time behaviour and displays with graphics.

Based on computer haptics, multimedia haptics brought the sense of touch into applications and integrated it with various media elements. The extra sensory modality greatly enriched the interactions between human and virtual environments as we can now communicate not only with sight and sound but also by touch. With their advantage in offering a physical interaction, haptic technologies have been widely used in many different *virtual-reality* (VR) fields.

Since haptics was originally utilized in robotics, there are abundant haptic applications that help human operators to perform a series of delicate manipulations using mechanical devices. With communication technology, such manipulation can be done teleoperation, which may significantly enhance human operators' safety in dangerous environments, such as combat, radiative areas, natural disasters and etc.. By reproducing a realistic physical contact through simulated tactile or force feedback, the human operators are able to have a better interaction experience as they can actually touch the object (see Figure 1.3).

Another field where haptic technologies are widely used is gaming. Gaming systems communicate with players using diverse sensory information, such as video, audio and, of course, touch. Vibration is a popular tactile feedback which was generated in controller(s)

⁷<http://robotik.dfki-bremen.de/en/research/robot-systems/exoskeleton-passive.html>

⁸<http://www.xbox.com/en-ca/>

⁹<http://www.immersion.com/haptics-technology/haptics-in-use/gaming.html>

¹⁰<http://www.novint.com/>



Figure 1.4: Haptic feedback interfaces or devices for gaming.

or joystick(s) in response to physical contact in the virtual world. Examples are shown in Figure 1.4a and Figure 1.4b. Some specific haptic devices promoted physical interaction from unidirectional sources, like vibration, to active haptics that receive and act simultaneously. Novint technologies, Inc., created an immersive communication device that reads a player's position in 3 DOFs and outputs force feedback in order to simulate physical contact, for example, the recoil caused by gunfire (see Figure 1.4c).



Figure 1.5: A surgeon student is practicing operation on a surgical training system.¹¹

Multimedia haptics is fundamental to training applications because the integrated sensory information of the system creates a realistic scenario in a virtual environment. For some surgeries, where surgeons or operators rely highly on the delicate resultant force from tool-tissue interactions and must require abundant training experience before an actual op-

¹¹<http://www.prweb.com/releases/2010/09/prweb4479024.htm>

eration [8], a haptic-based simulation and training system provides a cost-effective solution where the practice is rendered virtually without danger to patients. Illustrated by Liu *et al.*, the training system properly models the tissue’s mechanical behaviour, renders the tool-tissue interaction force and displaying it on haptic interfaces or devices [9]. Figure 1.5 demonstrates a surgical training system that applies haptic technologies.

Besides the samples stated earlier, many multimedia applications or systems integrate haptics for interaction diversity. E-Commerce, education, arts and design are the fields where haptics are currently in use or may be applied in the near future.

1.1.2 Human Head Stiffness Rendering

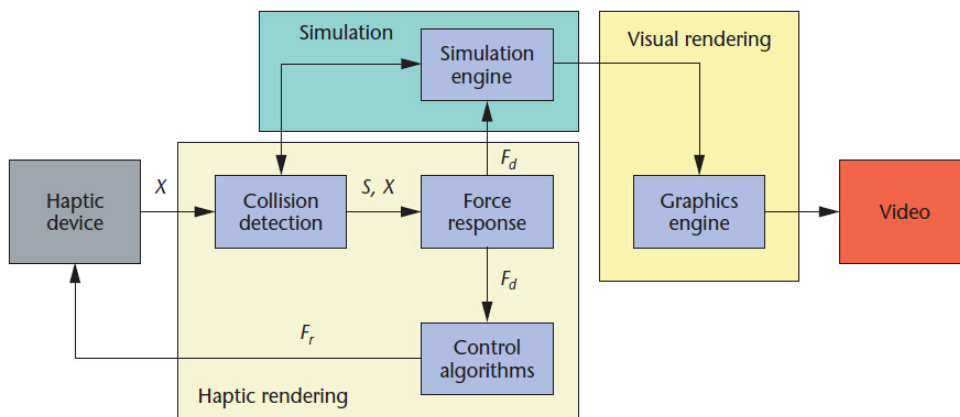


Figure 1.6: Blocks of haptic rendering[1].

In haptic rendering, the *force-response* is the core, as it directly associates with the interaction between avatar (a virtual object representing the haptic interface of the real world in the virtual environment) and virtual objects (see Figure 1.6 from Salisbury *et al.* [1]). With the positions of avatars and virtual objects, the resultant force can then be computed from a physical model simulating the interaction physically. The physical model integrates all essential elements that are used in rendering force feedback, including gravity, material and mechanics. Among the aforementioned elements, material is a crucial set of properties that describe the virtual object’s behaviour in response to different loadings, such as stresses and temperature. Stiffness is defined as resisting deformation when reacting to a force applied externally [10], and it is a similar concept to force-response as both of them describe the relation between interaction (deformation) and force. Therefore, rendering the proper stiffness of a virtual object is meaningful in haptic rendering for a realistic physical force feedback.

With the significance of stiffness in determining a relation between deformation and force, rendering the stiffness of a human head has a major role in providing realistic haptic feedback between a human operator and a touchable human avatar (proposed by professor

¹²<http://doctorstock.photoshelter.com/image/I0000yIPL67qhch0>



Figure 1.7: Structures of a human head in anatomy. ¹²

El Saddik and Multimedia Computing Research Laboratories (MCRLab), University of Ottawa [11]) created in the virtual environment. Head stiffness rendering for multimedia communication is a specific research topic that is challenging due to its complexity in both anatomic structure and geometric shape. As the human head has a sophisticated multi-layered structure composed of different types of tissue material, including skin, muscle, soft tissue and bone, its stiffness is highly difficult to be properly rendered for realistic interaction (see Figure 1.7). Rendering stiffness as a haptic property onto a 3D human head polygon mesh is a very specific research topic, while, to the best of my knowledge, there is not closely related work. In similar research, it is concerned with human head soft tissue material behaviour modeling from the perspective of biomedicine study, surgical simulation and even facial surgery prediction. In those studies, there are two approaches in order to create a haptic rendering for physical simulation: *Haptic Information Integration* and *Deformation Simulation*. In the former approach, the stiffness coefficients are simply defined as a constant value and integrated for easy interaction force computation using Hooke's Law [12]. However, the other one computes the force value with a series of complicated mathematic calculations which is able to model a more delicate material behaviour of a virtual object in a more physically realistic manner.

In terms of rendering the stiffness of a human head model, neither of the mentioned rendering approaches offer a satisfying solution due to several reasons. For the haptic information integration approach, the problems are as follows:

- **Material simplicity.** Due to the sophisticated anatomic structure, the material behaviour in a human head cannot be simply described using a set of constant values

as stiffness coefficient(s). Even though a force value can be computed with an user-defined coefficient value, the coefficients assignment could be tedious and the results from computation may lack accuracy.

- **Material linearity.** The assigned coefficients only represent the linearity in the behaviour where its deformation is proportional to the external force loading(s). As consequence, the effects caused by nonlinear terms of human soft tissue, such as viscoelasticity and hyperelasticity, are exclusive, and the inaction force resulting from interaction is not rendered properly.
- **Geometric structure simplicity.** The one-dimensional string structure modeled in Hooke's law is inadequate for modeling the behaviour of a virtual object with 3 DOFs.

Stiffness rendering procedures, which apply a complicated mathematical framework for deformation simulation, have their own defects in these aspects:

- **Medical image data requirements.** Most of the haptic applications associated with tool-tissue interaction force rendering require specific medical image data, such as CT scan, X-radiation, Magnetic Resonance Imaging (MRI), etc., in order to obtain accurate simulation results.
- **Computation expenses.** Generally the rendering process of nonlinear material is highly computation-expensive (with a time complexity as $O(n^2)$) [13] and is challenging to produce results in real-time.

1.2 Objective and Contribution

In this work, we propose a hybrid method to render the appropriate stiffness property on a human head polygon mesh that combines the popular haptic rendering approaches: to study the sophisticated deformation behaviour of a deformable object, then interpret and render this behaviour as the resultant stiffness property on the individual's head mesh. The stiffness property is estimated from a registered and shape-adapted skull template mesh as a reference, and modeled from soft tissue's deformation behaviour in a nonlinear Finite Element Method (FEM) framework. Our method consists of different procedures, including: facial landmark detection; model registration using Iterative Closest Point (ICP) technique; adaptive shape modification processed with a modified Weighted Free-Form Deformation (FFD); and FEM Simulation. After the stiffness property is rendered on a head polygon mesh, we perform a user study by inviting participants to experience the haptic feedback rendered from our results. According to participants' feedback, the head polygon mesh's stiffness property is properly rendered as it satisfies their expectation.

The contributes of this thesis are:

- We estimate the stiffness on a head polygon mesh by registering a skull template model to an individual’s head model for its supporting function to face structure.
- We design an approximation procedure that enables a skull template mesh adapt to diverse users’ head mesh through shape modification.
- We define the stiffness property using the deformation behaviour of soft tissue modeled with FEM simulations. The soft tissue is processed as a deformable medium filling the space between the head mesh and the shaped-modified skull mesh.

1.3 Thesis Organization

- **Chapter 2:** Here we provide a review of the literature of all related works, including facial landmark detection, model registration, shape modification, deformation behaviour modeling and haptics rendering in recent years.
- **Chapter 3:** In this chapter we mainly discuss the method of modeling the deformations of a 3D object. The basic concepts of continuum mechanics and the Finite Element Method are introduced.
- **Chapter 4:** The details in our proposed stiffness rendering method are elaborated in this chapter, which contains several procedures, such as *Landmarks Features Detection and Models Registration*, *Adaptive Skull Approximation*, and *Stiffness Property Rendering*. The implementations and results from experiments will be discussed in Chapter 5.
- **Chapter 5:** This chapter shows the results from Chapter 4 and details the experimental setup of the FEM simulation procedure. Furthermore, a discussion of result analysis is also covered.
- **Chapter 6:** Following to the experimental and simulation results, we implement a human head haptics interaction system. The details in implementation is described here, including system configuration, system architecture and introduction of each component. In addition, we evaluate the system by performing a user study. The method and data analysis will be discussed here as well.
- **Chapter 7:** We summarize the research work and also discuss future works that could further contribute to this field.

Chapter 2

Related Work

In order to understand the approach to render the proper stiffness property on a human head polygon mesh estimated from its reference skull model, this chapter mainly contains the works from multiple research domains: *Facial Landmark Detection*, *Models Registration*, *Shape Modification*, and *Deformation Behaviour Modeling*. Sections 2.1 and 2.2 give a brief introduction on the previous works of human facial landmark detection and anatomy registration. Section 2.3 discusses the shape modification techniques associated with our proposed skull-shape-adaption procedure. Section 2.4 introduces the principle of stiffness rendering from the perspective of *deformation modeling*, including the principle, merits and demerits of the existing deformable object modeling techniques. Section 2.5 discusses the previous works on haptic rendering in interactive virtual reality research.

2.1 Facial Landmark Detection on a 3D Point Cloud

3D facial landmark detection is emerging research that is not only capable of locating the positions of facial features for people’s identity in a three-dimensional structure, but also overcomes the limitations of common 2D detection techniques, such as illumination, expression and pose variations [14]. Based on the input image type, the detection technique uses a combination of color and depth information data [15, 16], or depth information data only [17, 18]. By adding depth information from points in a scene to the sensor device, range image extends a traditional 2D image to 3D. Segundo *et al.* present an automatic facial landmark detection method to locate a group of features (nose tip, nose corners and eye corners) on a range image [19]. However, a range image’s presentation of a 3D structure is incomplete because the depth information of the scene behind the sensor cannot be described. A 3D point cloud is a complete three-dimensional structure which can fully present the object’s geometric information in Cartesian coordinates. Since the acquisition data is a 3D polygon mesh instead of range image, we modify the approach presented in [19] and detect the facial features on a 3D point cloud.

¹³<http://www.clementcreusot.com/phd/>

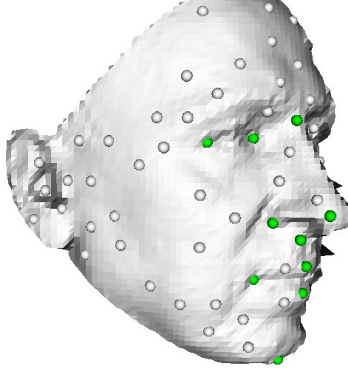


Figure 2.1: A sample 3D facial landmark detection. ¹³

2.2 Models Registration

Models registration involves merging multiple (two or more) individual datasets into a global model [20]. Besl *et al.* propose the ICP (Iterative Closest Point) algorithm for accurate and efficient registration of 3D shapes [21], which is fundamental in most registration methods. The main idea of ICP is to extract corresponding key points on both datasets and iteratively compute the transformation that minimizes the distance between the correspondences. The correspondences have to be selected from overlapping areas on datasets. In terms of registration of human anatomic structure, Koch *et al.* register a CT volume data of a head to reference from *Visible Human Data SetTM* using reference points for facial surgery planning [22]. In this work, we propose a semantic alignment method to register the head and template skull objects in which the estimated transformation is computed from a group of valid but shape-irrelevant facial features as correspondences to solve the issue of missing overlapping areas.

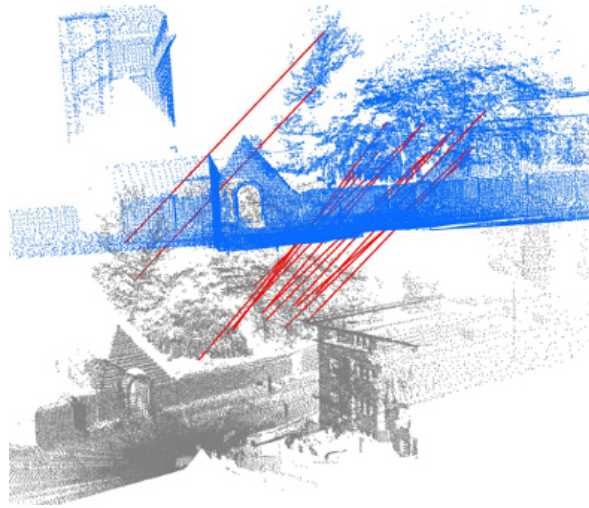


Figure 2.2: A sample of 3D models registration. ¹⁴

2.3 Shape Modification

Sederberg and Parry introduce a shape modification method, Free-Form Deformation (FFD), which deforms a rigid object in a free form manner [23]. A set of control points lattices defines a solid which encloses the object for process, and any change over the control points modifies the shape of both solid and the object (see Figure 2.3). Based on that, there are abundant works extended from regular FFD in different ways, such as parametric volume type [24, 25], control lattice topology [26, 27], and control point manipulation [28, 29]. Song and Yang enhance FFD by deforming the rigid object with weighted T-spline volumes that provide more flexibility in both control point topology and manipulation [30]. In order to properly modify the subtle shapes of a human skull and maintain its integrity in a whole, we designed an adaptive shape modification approach that hierarchically deforms the template skull mesh in different regions. Each region is processed using the weighted FFD with weight coefficients.

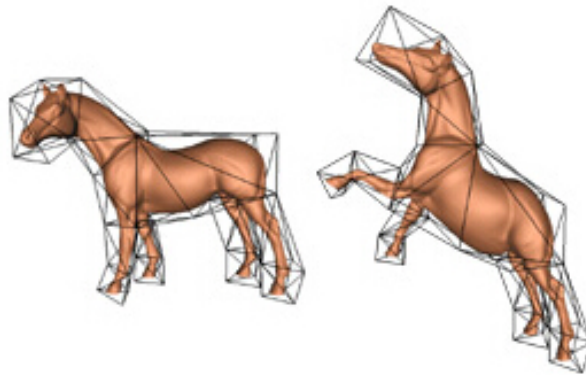


Figure 2.3: Deforming a rigid object by control point manipulation. ¹⁵

2.4 Stiffness Rendering

As stated earlier, the human head’s stiffness property is very difficult to describe directly, and hence it is often handled by material behaviour modeling of facial soft tissue. Many similar techniques are applied in simulating a haptic feedback on soft tissue as deformable object. Wang *et al.* introduce an innovative method in modeling organ’s deformation with a sphere-tree representation to provide an efficient and stable simulation in dental operations [31]. Jeon *et al.* propose an algorithm for providing realistic force augmentations simulating a breast tumor palpation scenario [32]. In the aspect of deformation modeling, Nealen *et al.* survey that the *Particle System* and the *Continuum Model* are the two most popular methods [33].

¹⁴<http://docs.pointclouds.org/trunk/groupregistration.html>

¹⁵<http://www.cs.rice.edu/~jwarren/research/>

2.4.1 Particle system

The particle system, also referred to as the mass-spring system (MSS), is a simple deformable model whose structure is discrete and is composed of a series of points with non-negligible mass, connected as a unity by a network of massless springs [34, 35, 36]. For each spring, the stiffness coefficient is a constant value and describes the force and deformation relation in one dimension. The structure is illustrated in Figure 2.1 by Nikolaev [37]. With its capability for modeling large deformation, MSS have been extensively applied in virtual object modeling, such as cloth simulation. More works in similar scenarios could be found in [38, 39].

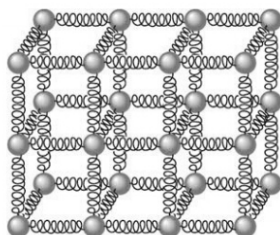


Figure 2.4: Mass-spring system structure. ¹⁶

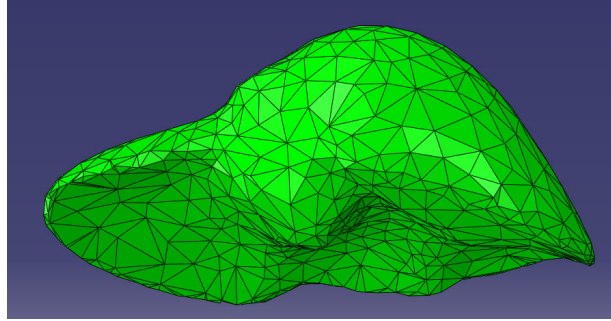
MSS's simplicity in structure makes the implementation simple, and hence the deformation can be simulated in real-time. Gao *et al.* render the real-time haptic feedback of simple virtual objects in a web application using the MSS [40]. In a scenario simulating human soft tissue material behaviour, the MSS is popular in rendering haptic feedback in many biomechanical applications, such as modeling liver in [41] (see Figure 2.5) and muscle in [42]. In most haptic rendering applications, the object's stiffness can only be described linearly because the spring property is assigned with constants.

However, more and more research is dedicated to modeling soft tissue's nonlinear behaviour using MMS. Koch *et al.* propose a multi-scale spring structure where each spring is composed by a set of smaller springs connecting in sequence [22]. Any spring i containing a discrete set of n sub-springs with their individual lengths l_i^1, \dots, l_i^n and stiffness coefficients k_i^1, \dots, k_i^n . The resultant k_i can be computed through line integration from 1 to n based on detected interaction. Xu *et al.* model nonlinear deformation of MMS using a strain-energy function [43]

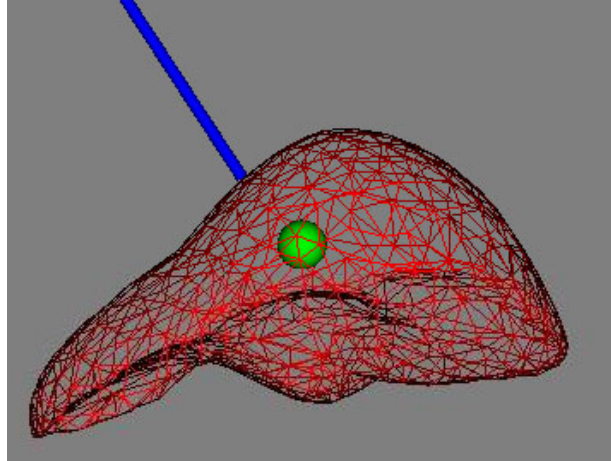
Despite of its advantage in real-time simulation, the particle system has its limitations in physical behaviour modeling accuracy for two reasons: 1) relying on the spring network's resolution and topology and 2) lacking systematic description of spring constants.

¹⁶<http://cowboyprogramming.com/2007/01/05/blob-physics/>

¹⁷<http://his.anthropomatik.kit.edu/english/176.php>



(a)



(b)

Figure 2.5: Liver modeled in a mass-spring system for haptic rendering; (a) is the preoperative model and (b) is the intraoperative one. ¹⁷

2.4.2 Continuum models

Deformable objects modeling in a physical manner is widely applied in many simulation applications. The continuum models process a virtual deformable object as a continuous mass, rather than discrete particles. The actual material properties can be integrated into the continuum model, making simulation results more accurate. Therefore, continuum model simulation techniques have been widely used in applications of virtual reality where delicate physical modeling is demanded. Nealen *et al.* [33] and Meier *et al.* [44] provide discussions of physically-based deformable objects in computer graphics and their applications in surgical simulation, respectively. A virtual object for continuum modeling does not have to be solid, but could also be fluid or gaseous, which makes the simulation available in different scenarios.

A regular polygon mesh normally structures a 3D model object's geometric shape on surface by a set of facets connecting with each other. Each facet, either a triangle or a quadrilateral depending on its shape, is a planar surface in a 3D environment. Therefore, a polygon mesh is analogous to a hollow 3D object with nothing but its surface. While

¹⁸<https://www.cst.com/Products/CSTmws/FrequencyDomainSolver>

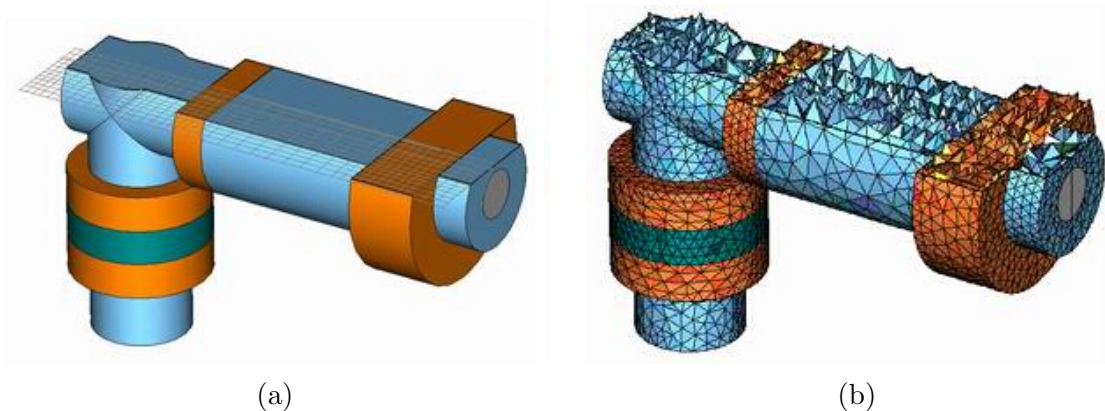


Figure 2.6: Volume mesh of a continuum model; (a) is the surface of a volume mesh, and (b) is the cut plane of the volume mesh¹⁸.

volume mesh comprises a group of continuously connected elements that make it capable of representing an object's interior volume property, hence can be used for continuum mechanics modeling. When the object for modeling is solid, there are several types of geometric element shape and four of them are mostly used in continuum modeling: tetrahedron, pyramids, prism, and hexahedron (see Figure 2.7). The tetrahedron is often the default element type in 3D volume meshing for its applicability and refinement ability. Any 3D object, no matter its shape or topology, can be meshed with tetrahedrons. Besides, it is the only type of element capable of refinement and adaption during meshing.

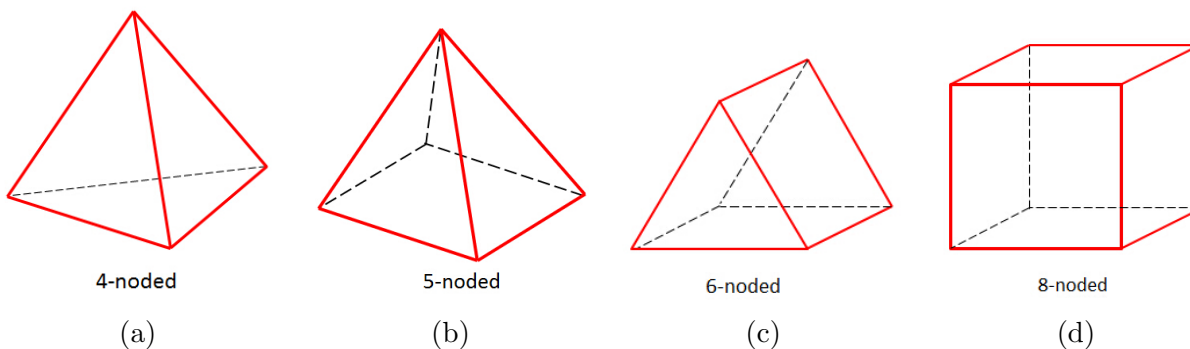


Figure 2.7: Volume mesh element types; (a) is a tetrahedral element; (b) is a pyramid element; (c) is a prism element, and (d) is a hexahedral element.

Finite Element Method (FEM)

In a continuum model, the deformation information is described by a set of *Partial Differential Equations* (PDE) which are impossible to be solved numerically. In order to solve this problem, many mathematic frameworks, including the *Finite Element Method* (FEM), *Finite Volume Method* (FVM) and *Boundary Element Method* (BEM), provide solutions by describing the physical rules suitable for solid objects. FEM is the most popular method as

it is more frequently applied in physical modeling and simulation. Firstly, FEM discretizes the complete solid object into a finite number of sub-domains, also referred to as *elements*. Then, for each element, it solves a set of equations contained by that element describing its physical states of stress, strain and displacement. When results of all elements are obtained, FEM approximates their sum as the solutions of the whole domain.

Linear FEM. The linear FEM specifies linearity in both geometry and material of the continuum model. Geometrically, a linearized relation of Cauchy's strain is applied instead of a Green strain. Such alteration enhances computation efficiency but is limited to small deformations, especially of a rigid body's rotations. In order to overcome poor performance of linear FEM in handling large deformation, Mller and Gross [45] propose a co-rotational method called *Co-Rotational FEM* that separates the rotation component from deformation of each element in the reference configuration and rotates back to current configuration. Hence, the computation of forces is based on the non-rotated reference.

Regarding material, the constitutive law relating strain and stress of the virtual object in most applications is described linearly due to implementation simplicity and low computation cost. In the case of soft tissue behaviour modeling, the material's viscoelasticity and hyperelasticity must be modeled using a nonlinear constitutive law.

Nonlinear FEM. In contrast to linear FEM, if both linearity conditions mentioned previously are not satisfied, the deformable object shall be modeled nonlinearly. Zhuang and Canny introduce an approach to simulate a 3D object's large global deformations in real time using a quadratic nonlinear strain [46]. Furthermore, they apply an explicit integration scheme to avoid costly computation of nonlinear systems in order to improve efficiency. Debonne *et al.* present an adaptive method to animate the viscoelasticity of deformable objects dynamically [47]. In their approach, the computational resources are concentrated, as the space and time sampling has been locally adapted to current local deformation. Wu *et al.* apply nonlinear FEM using mesh adapting to obtain sufficient details in local deformation that minimize unnecessary computation [48].

In terms of material nonlinearity, there have been many works studying a nonlinear strain-stress relationship in constitutive law. Temam and Miranville introduce the principle of continuum mechanics and discuss the mathematical model in nonlinear constitutive laws [49]. For nonlinear materials, energy is stored by the mechanics system during deformations as *strain energy* [50] and it can be described using *strain energy function*. The function relates the Cauchy-Green tensor and strain energy density, and its invariants can further be used in describing the stress tensor. Boonvisut *et al.* estimate the soft tissue mechanical parameters with a nonlinear finite element scheme [51]. Teran *et al.* simulate the contraction of skeletal muscle tissue on a hyper-elastic constitutive model [52]. Yan *et al.* [53] and Chai *et al.* [54] present deformation modeling of soft tissue solved by FEM with material nonlinearity in surgical simulations.

2.5 Haptic rendering

Haptic rendering is the core component in any interactive physical simulation, as it properly detects the interaction between haptic tool and virtual objects, computes the resultant force value and reproduces haptic feedback through haptic interfaces and devices. As Illustrated by Srinivasan and Basdogan, haptic rendering techniques could be classified based on interaction types as: *point interaction* and *ray interaction* [6].



Figure 2.8: Haptic interaction types: (a) is the point-type interaction ¹⁹ and (b) is the ray-type ²⁰.

For point-type haptic interactions, the end point of the haptic tool, also referred to as *haptic interface point* (HIP), is the only part interacting with virtual objects. In a collision event, HIP inside the object is captured, and the depth of penetration (distance between HIP and its closest point on object's surface) will be used in computing interaction force. Point interaction is appropriate for modeling a 3 DOFs haptic rendering, since the torque is not necessary in this process (see Figure 2.8a). Ray-type interactions are more complicated because the haptic tool is represented as a finite ray and its orientation is required in the process, which extends manipulation to more DOFs. Moreover, the collision detections obtain the intersection point between the ray and object's surface and hence are available for simulating more delicate manipulation, such as dissection, rather than "poking" (see Figure 2.8b).

While in scenarios simulating non-penetration constraints or objects with thin or complex shapes, the haptic rendering process could be challenging using HIP. To overcome these limits, Zilles and Salisbury propose haptic rendering algorithms to generate realistic interaction forces of objects processed as a rigid polyhedra, called a *god-object* [55]. A god-object represents HIP's virtual location constrained on an object's surface, and the location is often chosen to be the surface point with the minimum distance from HIP to god-object.

Initially, force feedback was mainly the force generated from the penetration depth resulting from tool-object interaction. With the development in haptic rendering technology, more complicated haptic properties, including material stiffness, texture and friction,

¹⁹<http://watchongadget.net/speed-master-wins-a-novint-falcon>

²⁰<http://www.geomagic.com/en/products/phantom-desktop/overview/>

have been taken into account in recent haptic feedback applications. Coulomb and viscous friction have been modeled in haptic interactions in early research works [56, 57].

Chapter 3

Background Information of Continuum Model and Finite Element Method

3.1 Continuum Model

Derived from mechanics, continuum mechanics analyses a material's kinematics and mechanical behaviour as a continuous mass other than a series of discrete particles. Different continuum models, like plasticity, viscosity, elasticity and etc., can be applied when modeling different material behaviour.

For general deformation modeling, where linear elastic material is specified for simplicity, the deformable object's displacement is linearly proportional to the applied force under small force and deformation conditions. Such deformation behaviour can be described with Hooke's Law for a linear spring as:

$$\mathbf{f} = \mathbf{K}\mathbf{u} \tag{3.1}$$

where \mathbf{f} is the applied force on deformable object, \mathbf{K} is the constant spring stiffness, and \mathbf{u} is the spring's elongation or compression at equilibrium status. In a continuum model, we categorize using the four associated physical status of displacement, strain, stress, and force, and material behaviour is described by the relationship between specific status (shown in Figure 3.1). The relationships consist of:

- Kinematic laws relating displacement, or deformation, of vertices on deformable object and strain;
- Constitutive law relating strain and stress;
- Equilibrium equations relating stress and the applied external force; and
- Stiffness description relating displacement and applied force.

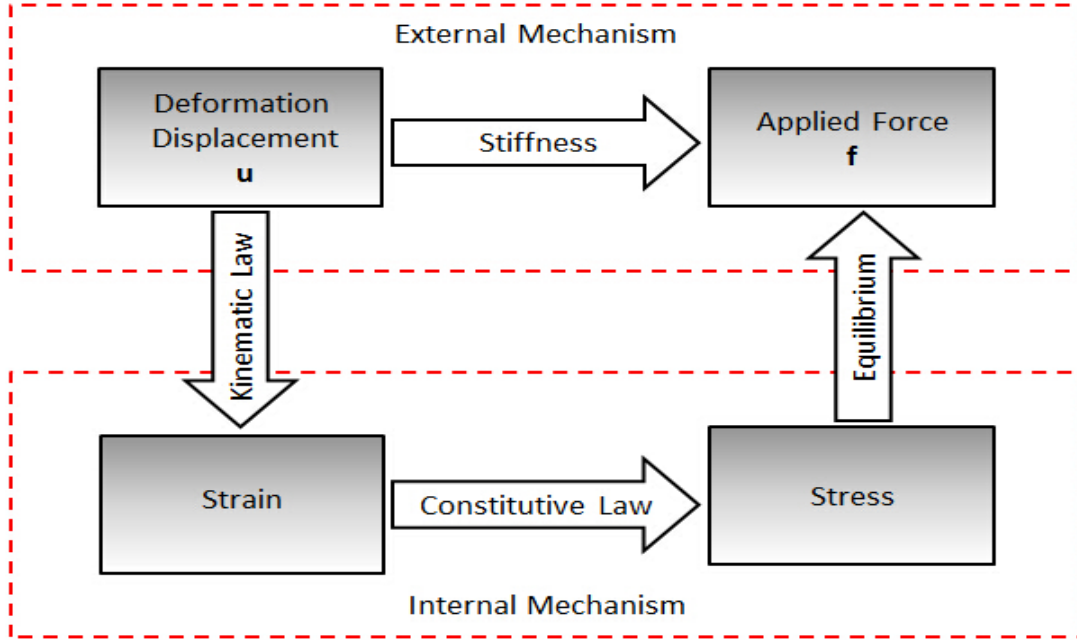


Figure 3.1: Relationships between the physical status in continuum model

In contrast to the general form of Hooke's Law, where stiffness is described as a 1-D spring, we need to extend the \mathbf{K} from a real number to a matrix of real numbers to express the mathematical description of deformation because the deformable object we are modeling is a 3D object with higher dimensions.

3.1.1 Strain

Strain measures the deformation in a quantitative manner. Strain can be analogously interpreted as a ratio of elongation compared to the original length in a 1-D spring case. However, for a 3D object, ϵ is the strain tensor of a deformable object and is presented with a 3×3 symmetric matrix as

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix} \quad (3.2)$$

where

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_{k=1}^3 \frac{\partial u_k}{\partial u_i} \cdot \frac{\partial u_k}{\partial x_j} \right) \quad (3.3)$$

The matrix in Equation 3.2 is also known as the Green strain tensor, where x, y, z are the Cartesian axes, i, j, k are the coordinates of each degree of freedom in a 3D space, \mathbf{u} is the interpolated element displacement vector, and \mathbf{x} is the coordinate of a point on deformable object. The Green strain tensor is derived from the deformation gradient as:

$$\epsilon = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) \quad (3.4)$$

where

$$\mathbf{F}^T \cdot \mathbf{F} = (\mathbf{R} \cdot \mathbf{U})^T \cdot (\mathbf{R} \cdot \mathbf{U}) = \mathbf{U}^T \cdot \mathbf{R}^T \cdot \mathbf{R} \cdot \mathbf{U} = \mathbf{U}^T \cdot \mathbf{U} \quad (3.5)$$

this proves Green strain tensor only depends on deformation \mathbf{U} while being completely independent of rigid body rotation \mathbf{R} . The Green strain tensor in Equation 3.3 is a general form which is capable of handling both linear and nonlinear situations; however, it can be further simplified to a linear Cauchy tensor only when the object's material property is processed linearly and deformation is small. When its nonlinear term is negligible, Equation 3.3 can be written as:

$$\epsilon = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.6)$$

3.1.2 Stress

When a deformation occurs on a deformable solid object, it always produces internal stress on every node inside the continuous material. *Stress* σ is used to express distribution of the generated internal forces and their influence on each other in continuum mechanics. For a 1-D case such as spring, stress can be represented as the reaction force generated from spring's elongation or compression. Similar to strain, for each vertex in the material body, its stress is presented as a symmetric 3×3 matrix:

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \sigma_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \sigma_{zz} \end{bmatrix} \quad (3.7)$$

where x, y, z are the Cartesian axes and σ_{ij} is the stress on the perpendicular plane i along j direction. The diagonal entries are orthogonal normal stresses, and the other entries are shear stresses and often written as γ . Normal stress describes the stress acting perpendicular to the cross-section, while the shear acts tangential to the surface.

3.1.3 Constitutive Law

Associating strain with stress in a continuum model, the constitutive law models the actual material behaviour with its definition of the response of the deformable object to external

stimuli. In most of the cases when a linear elastic material behaviour is to be modeled, the strain and stress are related using Hooke's Law with a linear function:

$$\sigma = \mathbf{K} \cdot \epsilon \quad (3.8)$$

where \mathbf{K} is a 6×6 matrix and contains only two material properties, and Young's modulus E and Poisson's ratio μ describe a material's stiffness and compressiveness respectively. As a consequence, \mathbf{K} can be fully expressed as follows:

$$\mathbf{K} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 - 2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - 2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 - 2\nu \end{bmatrix} \quad (3.9)$$

Since the Young's modulus E and Poisson's ratio μ are two constants specified as input, all values in \mathbf{K} are constants and it can be pre-computed in advance to reduce the computation work needed during simulation, which significantly enhances efficiency in most real-time applications. However, the values in \mathbf{K} of a nonlinear material fluctuate in response to current deformation. Hence, \mathbf{K} must be recalculated in simulation. The details of nonlinear constitutive law will be explained in Section 3.3

3.2 Finite Element Method

The concept of FEM has been briefly introduced earlier in Section 2.4.2, while the principle and specific procedures will be discussed here. While the behaviour of a virtual object is simulated as a unity, each element contains its local \mathbf{K}_e represented in Equation 3.8 and is independent of others. Then, all elements from continuous domains are assembled into a global system and computed for final calculation. Figure 3.2 gives a brief introduction of the FEM framework that we use to model the deformation behaviour of a human head model.

3.2.1 Discretization

Firstly, a deformable object is discretized into a set of m elements e_1, e_2, \dots, e_m , and these elements are where the displacement and force are approximated. However, because nodes are where the force and displacement being evaluated, we need to express force and displacement values within elements to a set of N nodes, n_1, n_2, \dots, n_N properly with interpolation. Interpolation, also known as shape function ϕ_i , reconstructs the nodal displacement u_i from discrete samples to a continuous displacement $u(\mathbf{x})$ for any arbitrary point within an element. The interpolation function depends on the type of volumetric mesh, such as tetrahedral or hexahedral mesh.

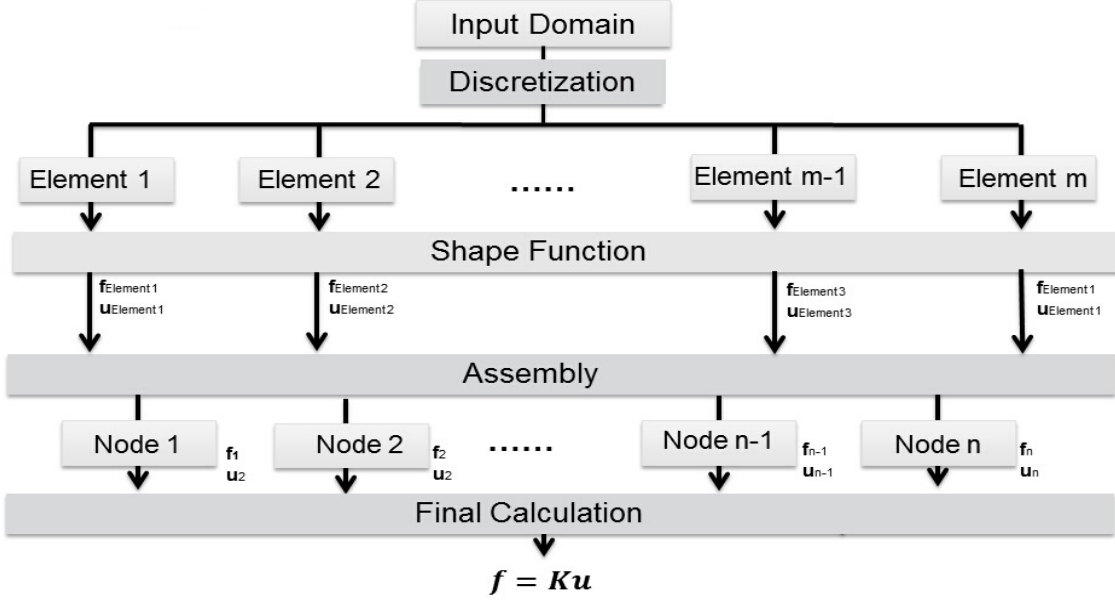


Figure 3.2: FEM framework pipeline

$$u(\mathbf{x}) = \sum_{i=1}^l \phi_i(\mathbf{x}) \mathbf{u}_i \quad (3.10)$$

where l is the size of the nodes for an element, and therefore shape function ϕ_i depends on element type of the volumetric mesh. As a tetrahedron, where $l = 4$, is the element type in our research topic, the resultant $u(\mathbf{x})$ can be taken as displacement of the element's barycentric coordinates in this case.

In a FEM domain where its elements and material are described linearly, the relationships between strain ϵ and deformation \mathbf{u} , stress σ and strain ϵ , can both be interpreted through linear functions. With the \mathbf{u}_e which comprises collected nodal displacement vectors on an element and computed by shape function, the kinematic law relating strain ϵ and displacement \mathbf{u} in an element can be expressed with geometric linearity as

$$\epsilon_e = \mathbf{B}_e \cdot \mathbf{u}_e \quad (3.11)$$

where \mathbf{B}_e is a constant matrix with size of $[6 \times 3l]$. Strain ϵ_e can be used in representing the stress σ_e of an element by constitutive law, with material linearity as

$$\sigma_e = \mathbf{K}_e \cdot \epsilon_e \quad (3.12)$$

where \mathbf{K}_e is already defined in Equation 3.9. Similarly to \mathbf{K} , \mathbf{K}_e is a $[6 \times 6]$ constant matrix where its values are related to Young's Modulus E and Poisson's ratio μ only. Thus, the displacement and force over an element can be related with a function as:

$$\mathbf{f}_e = \mathbf{K}_e \cdot \mathbf{u}_e \quad (3.13)$$

where $\mathbf{K}_e = V_e \mathbf{B}_e^T \mathbf{C}_e \mathbf{B}_e \in R^{3l \times 3l}$, l is the number of nodes on an element, V_e is the element volume, for a tetrahedral element \mathbf{B}_e is a 6×12 matrix in linear deformation function of $\epsilon_e = \mathbf{B}_e \cdot \mathbf{u}_e$. With shape function, Equation 3.12 can be written in discretized form with nodal force and displacement in vectors.

3.2.2 Assembly

Since kinematic and constitutive law of every element has been described in the previous procedure, we must take all elements into consideration in order to model the deformation behaviour of the object as a whole for final calculation. The assembly of elements must satisfy two conditions:

- The nodes' displacement must be the same as sharing node(s);
- The sum of all interior element forces contributing to a specific node must be equivalent to the force acting at that node externally.

Because the resultant displacement and forces of the entire domain are expressed over nodes in the global system, the assembly is a straightforward process by adding stiffness matrices associated with a particular node. This is numbered globally until a final $[3n \times 3n]$ global stiffness matrix is completed, where n is the size of nodes in the volumetric mesh. The global stiffness matrix is a sparse symmetric mesh, and each element inside describes the connection between the two corresponding nodes. As a consequence, the relation between elastic force and displacement vectors over the nodes in a global system can be expanded from Equation 3.1 to a 3-D format:

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}_{3n \times 1} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1n} \\ \mathbf{K}_{21} & \ddots & & \vdots \\ \vdots & & & \\ \mathbf{K}_{n1} & & & \mathbf{K}_{nn} \end{bmatrix}_{3n \times 3n} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}_{3n \times 1} \quad (3.14)$$

$$\mathbf{f}_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix}_{3 \times 1}, \mathbf{K}_{ij} = \begin{bmatrix} k_{ij11} & k_{ij12} & k_{ij13} \\ k_{ij21} & k_{ij22} & k_{ij23} \\ k_{ij31} & k_{ij32} & k_{ij33} \end{bmatrix}_{3 \times 3}, \mathbf{u}_i = \begin{bmatrix} u_{ix} \\ u_{iy} \\ u_{iz} \end{bmatrix}_{3 \times 1} \quad (3.15)$$

where n is the size of nodes on volume mesh, \mathbf{f}_i and \mathbf{u}_i are two 3-D vectors that describe the elastic force and displacement at a particular node N_i . The K_{ij} , an element in the global stiffness matrix, represents the stiffness effect between the nodes N_i and N_j . As shown in Equation 3.14, the relation between elastic force and displacement at a node is not simply evaluated by discrete units, but approximated with a sum of forces affected by connections between nodes.

3.3 Nonlinearity of the Finite Element Method

The continuum behaviour can be modeled linearly only when the two conditions, geometric and material linearity, are satisfied simultaneously. The geometric linearity, where the relation between strains ϵ and displacement \mathbf{u} is described in a linear function, is valid only when the \mathbf{u} is relatively small, especially the rotation. Otherwise, the large deformations would cause distortion in the simulation results. On the other hand, the material linearity represents deformable linear elasticity in the constitutive law, as the stresses σ and strains ϵ are correlated with a linear function. However, if any or both of the linearity conditions cannot be met, the deformable object's behaviour should be modeled by nonlinear FEM instead of the linear one.

Since human soft tissue is a hyperelastic material property, its constitutive law is no longer linear and must be described in a nonlinear manner. Thus, in this section, we focus on the nonlinear constitutive law of a hyperelastic material. Teran *et al.* [52] model the behaviour using a scalar function $W(G)$ that describes the strain and its energy at each point as a function. As the simulation context in our research topic is associated with soft tissue's passive property only, we make modifications based on [52] and represent the strain energy in the form as

$$W(I_1, I_2, \lambda, \mathbf{a}_0, \alpha) = F_1(I_1, I_2) + U(J) + F_2(\lambda) \quad (3.16)$$

where I_1 and I_2 are a strain's deviatoric isotropic invariants, λ is a strain invariant relating to transverse isotropy, \mathbf{a}_0 is the direction of barycenters from two connecting elements, F_1 is the isotropic term, $U(J)$ is the term representing incompressibility, and F_2 describes the passive behaviour. F_1 can be fully expanded as

$$F(I_1, I_2) = A \times I_1 + B \times I_2 \quad (3.17)$$

where A and B are two material constants, while I_1 and I_2 are written as

$$I_1 = \text{tr}(\mathbf{C}) \quad (3.18)$$

$$I_2 = \frac{1}{2((\text{tr}(\mathbf{C}))^2 - \text{tr}((\mathbf{C})^2))} \quad (3.19)$$

where $\text{tr}(\cdot)$ is the trace of (\cdot) and $\mathbf{C} = J^{-\frac{2}{3}} \mathbf{F}_T \mathbf{F}$.

$U(J) = K \ln(J)^2$ describes the volumetric response of incompressibility where J is the related volume, and the degree of incompressibility can be defined using the magnitude of bulk modulus K . In F_2 , the stretch's direction λ is adjusted in order to follow \mathbf{a}_0 , the barycenters' direction.

As a result, the stress in the deformable element can be written as

$$\begin{aligned} \sigma = p\mathbf{I} + \frac{2}{J}((W_1 + I_1W_2)\mathbf{B} - W_2\mathbf{B}^2 + W_\lambda\lambda^2\mathbf{a} \otimes \mathbf{a}) \\ - \frac{2}{3J}(W_1I_1 + 2W_2I_2 + W_\lambda\lambda^2)\mathbf{I} \end{aligned} \quad (3.20)$$

where $B = J^{-\frac{2}{3}}\mathbf{F}\mathbf{F}^T$, $W_i = \frac{\partial W}{\partial I_i}$, $\mathbf{a} = J^{-\frac{1}{3}}\frac{\mathbf{F}\mathbf{a}_0}{\lambda}$, $p = K\frac{\partial U}{\partial J}$ and \otimes is the tensor product manipulator.

Comparing the nonlinear FEM model with the linear models, the only difference is the nonlinearity in constitutive law as it specifies the hyperelasticity of soft tissue in material property. In terms of the simulation process, every procedure is the same as described in section 3.2 except for substituting the constant matrix with a nonlinear function in the description of stress-strain relation (Equation 3.12).

Chapter 4

Methodology

4.1 System Architecture

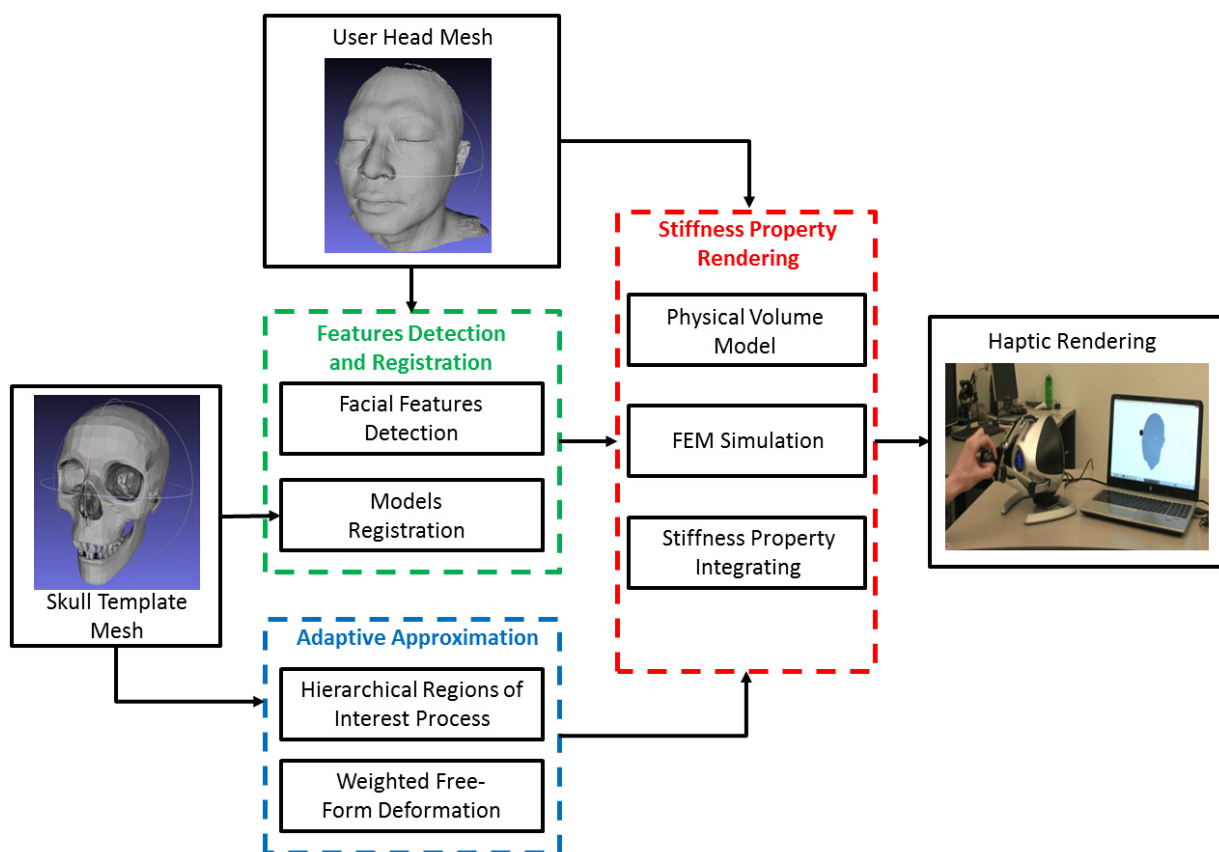


Figure 4.1: Overview of system architecture.

The proposed stiffness rendering approach is to integrate the haptic property (stiffness, in the scope of our thesis) onto an individual user's head polygon mesh that enables physical

interaction between a human operator and the virtual head via haptic rendering. Given a 3D head polygon mesh as input, the rendered stiffness property shall represent a nonlinear behaviour in deformation and is dependent on facial regions and tool-object intersection detected during haptic rendering. Figure 4.1 gives an overview of architecture of our rendering approach that contains 3 main components:

- *Landmark detection and models registration* properly locates a skull template mesh inside the head mesh, which is registered through facial features detected from both meshes.
- *Adaptive skull approximation* adjusts the geometric shape of a skull template mesh to head mesh on the outside via modification procedure.
- *Stiffness property rendering* renders the head model’s stiffness in two steps: 1) models the deformation behaviour of soft tissue material by FEM simulation, and 2) interprets the stiffness coefficients from deformation behaviour and integrate them onto the target user head polygon mesh.

4.2 Facial Landmark Detection and Models Registration

The first step is to register the template skull mesh to the individual head mesh. As the head and skull meshes cannot be aligned directly using a regular *Iterative Closest Point* (ICP), we propose a semantic registration method to align the two objects through a group of specified facial features instead of objects themselves. Those correspondences are independent from surface shape variance and hence overcome the problem of lacking overlapping areas in the regular ICP. The facial features on a head mesh are detected using landmark detection, and those on the skull mesh are generated from aesthetic proportions.

4.2.1 Facial Landmark Detection on a 3D Point Cloud

We apply the facial landmark detection technique introduced by Segundo [19] and make modifications to adapt that approach from a range image case to a 3D point cloud. The procedure is mainly processed in 3 steps: 1) identify surface type with curvature, 2) detect nose features (nose tip, nose corners) and 3) detect eye features (eye corners). The detailed procedures are explained as follows.

Surface Type Identification with Curvature

From the head model, the face region is the only part that contributes to landmark detection and must be segmented for analysis. We use a radius search method to segment face region, where the search center \mathbf{P}_{zmax} is the point with maximum z-value and radius R set

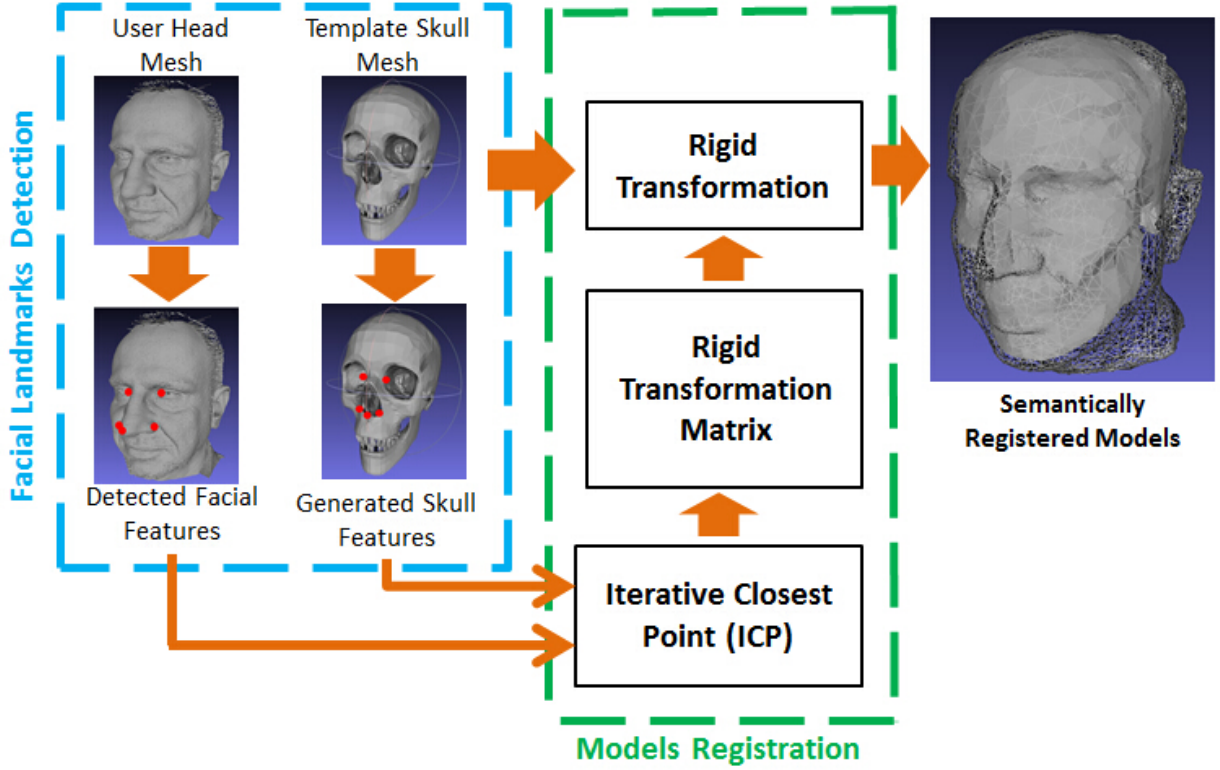


Figure 4.2: Illustration of facial landmark detection and models registration procedures.

to 9 *mm*, and points inside the search radius are identified as a face point. For every point on the isolated face region, we identify its surface type with curvature information. Firstly we create a quadric surface that fits a group of 50 neighbour points $\{(x_i, y_i, z_i)\}_{i=0}^{m=50}$ that minimizes the sum of squares of the offests of the points as

$$\|\delta\|^2 = \min \sum_{i=0}^m w_i [\mathbf{a}_i \varphi_i(x_i, y_i) - f(x_i, y_i)]^2 \quad (4.1)$$

where w is a set of weights, a is the estimator coefficients for the quadric surface, φ is the function space, and $f(x, y)$ is the input dataset that satisfies $z_i = f(x_i, y_i)$. By setting the weight always to be 1, and the function space as $\varphi = \text{span}\{y^2, y, x^2, x, 1\}$, the format of quadric surface function can be expressed as $z = f(x, y) = a_0 y^2 + a_1 y + a_2 x^2 + a_3 x + a_4$. Assuming the minimum value exists, then we make $\frac{\partial \sigma}{\partial x} = 0$, and the least square estimator can be computed by solving the equation below:

$$\mathbf{a} = (\varphi \varphi^T)^{-1} \varphi^T f(x, y) \quad (4.2)$$

When the quadric surface coefficients are estimated and the square difference offests satisfies the error threshold, like the example shown in Figure 4.3, now we are able to proceed in estimating the Gaussian curvature K and Mean curvature H . On a regular surface $p(x, y) = (x, y, z) = (x, y, f(x, y))$ in \mathbf{R}^3 , K is given in Equation 4.3 and H is given in Equation 4.4 respectively:

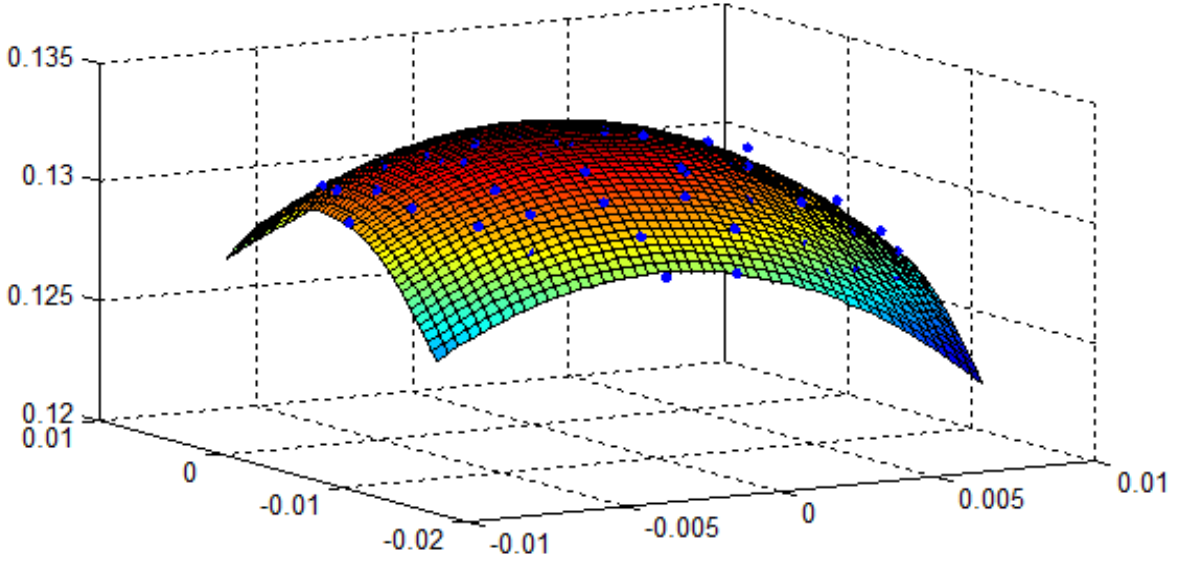


Figure 4.3: Example of a quadric surface with least square fitting generated from a 50-neighbour sample around nose tip on a 3D human head point cloud dataset. Points represent the input 50-neighbour.

$$K = \frac{eg - f^2}{EG - F^2} \quad (4.3)$$

$$H = \frac{eG - 2fE + gE}{2(EG - F^2)} \quad (4.4)$$

where E , F and G are coefficients of the first fundamental form while e , f and g are coefficients of the second fundamental form. Here we make $u = \frac{\partial f(x,y)}{\partial x}$ and $v = \frac{\partial f(x,y)}{\partial y}$, hence the first partial derivatives can be written as $p_x = (1, 0, u)$ and $p_y = (0, 1, v)$, second partial derivatives written as $P_{xx} = (0, 0, \frac{\partial u}{\partial x})$, $P_{xy} = (0, 0, \frac{\partial u}{\partial y})$, $P_{yx} = (0, 0, \frac{\partial v}{\partial x})$, and $P_{yy} = (0, 0, \frac{\partial v}{\partial y})$, and the normal vector of the surface $\mathbf{n} = \frac{1}{\sqrt{1+u^2+v^2}}(u, v, -1)$. K and H then can be estimated by substituting the values the equation, and by first, second derivatives and normal vector as: $E = 1 + u^2$, $F = u \times v$, $G = 1 + v^2$, $e = \frac{\frac{\partial u}{\partial x}}{\sqrt{1+u^2+v^2}}$, and $f = \frac{\frac{\partial u}{\partial y}}{\sqrt{1+u^2+v^2}} = \frac{\frac{\partial v}{\partial x}}{\sqrt{1+u^2+v^2}}$, and $g = \frac{\frac{\partial v}{\partial y}}{\sqrt{1+u^2+v^2}}$. The surface type can be classified using Table 4.1 proposed by Besl *et al* [58]. with the estimated K and H values.

Nose Features Detection

Since the head model is pre-processed and its frontal orientation aligned to z -direction in Cartesian coordinate systems, we use a straightforward nose tip locating approach that assumes the nose tip is the point \mathbf{P} whose z value is maximum. Once the position of

	$H < 0$	$H = 0$	$H > 0$
$K < 0$	Saddle Ridge	Minimal	Saddle Valley
$K < 0$	Ridge	Flat	Valley
$K < 0$	Peak	(None)	Pit

Table 4.1: Surface type classification table.

nose tip $\mathbf{P}_{NoseTip}(x_{NoseTip}, y_{NoseTip}, z_{NoseTip})$ is identified, we compute a x -projection that represents the points with the same y -coordinate value as $y_{NoseTip}$ to detect nose corners. On the horizontal profile curve (x -projection), we compute the gradient information and define positions of nose corners as gradient peaks on each side of the located nose tip.

Eye Features Detection

To detect eye features on a head point cloud, their positions are computed based on curvature information. Firstly, a y -projection is generated from the percentages of pit-curvature points of each collection of points whose y values are the same. The positions with the highest proportion of pit-like curvature are considered as eye features' y -coordinates on the computed projection.

Secondly, a different x -projection of curvature is generated to locate the eye corners' position on x -direction. For every point on the y -coordinate, we compute the percentage of pit curvature points of its 7 neighbour points filtered along with the y -direction. In the projection, the left eye corner is processed as the beginning of first peak while the right one is the end of the second peak.

4.2.2 Models Registration

As stated before, our proposed registration procedure is performed semantically as the two models, skull and head, are different objects that cannot be done with a regular *ICP*. As the facial landmarks on a head point cloud are already detected, the registration contains two steps, including *Skull Features Generation*, and *Semantic ICP Registration*.

Before generating the skull's facial features, we firstly scale the point cloud to an approximate size to the head point cloud. After trimming irrelevant parts, such as ears and neck, the scaling vector $\mathbf{S}(s_x, s_y, s_z)$ is computed while s_x , s_y , and s_z , are factors on each direction respectively as a ratio between the two objects' length. After being scaled, the corresponding positions of facial features on skull point cloud are created using aesthetic proportions as shown in Figure 4.4.

With two sets of corresponding points, $Head = \{Head_1, \dots, Head_5\}$ and $Skull = \{Skull_1, \dots, Skull_5\}$, we estimate the alignment information based on these specified semantic features using *ICP* procedure. Since the skull point cloud is the object to be

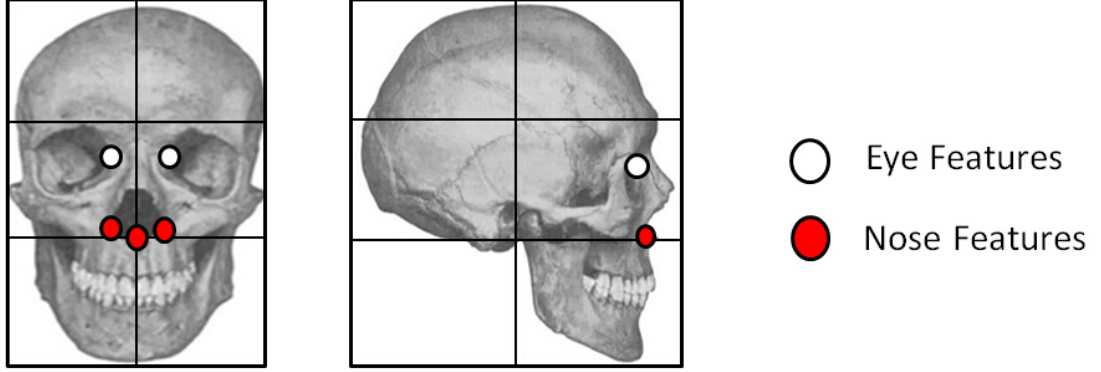


Figure 4.4: Aesthetic proportions of corresponding features on the skull model.

registered, the *skull* feature points are considered to be the source for transformation while the *head* features are kept fixed as the target. *ICP* computes the translation vector \mathbf{T} and rotation matrix \mathbf{R} that minimizes the sum of the squared error between the sets of features as

$$E(\mathbf{R}, \mathbf{T}) = \frac{1}{N_p} \sum_{N_p}^{i=1} \|\mathbf{x}_i - \mathbf{R}p_i - \mathbf{T}\|^2 \quad (4.5)$$

where $Head_i$ and $Skull_i$ are pairwise corresponding points and N is the number of pairs (here $N=5$). The computation is processed iteratively until the desired convergence criterion is met. The complete procedure of the semantic *ICP* is summarized in Figure 4.5.

When the resultant transformation matrix is estimated from *ICP* analysis, we transform (rotate and translate) the entire skull template point cloud accordingly with that matrix. In consequence, the skull object is aligned to the head and the geometry difference between their positions is minimized.

4.3 Adaptive Skull Model Approximation

As long as the head and skull meshes are registered, the skull template mesh's shape can then be modified in order to adapt an individual's head mesh. Our approximation is to minimize the shape difference by modifying the template skull mesh's shape using a Free-Form Deformation (FFD) technique. Because of the complexity of the shape and region dependency of the skull mesh, we propose a modified weighted FFD method which processes it on the Regions of Interest (ROI) in hierarchy. Figure 4.6 gives an overview of our adaptive approximation approach. The details of hierarchical ROI and weighted FFD are extended as follows.

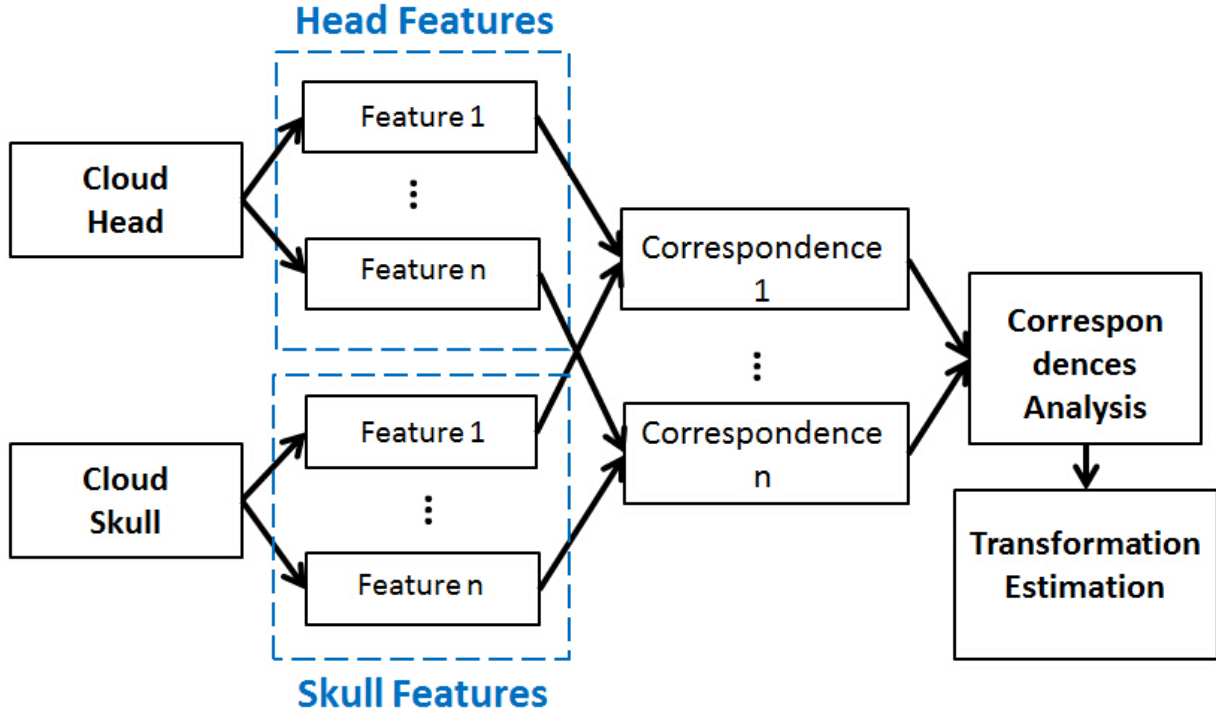


Figure 4.5: The pipeline of the semantic ICP registration approach.

4.3.1 Hierarchical ROI

Even though our approximation is achieved with FFD, the manipulation of control points, which practically specifies the deformation of a solid object, is a prerequisite to approximation of high quality. First of all, since the complete process of approximation is through self-adaption without manual operations, it makes automatic and precise control point manipulation essential in our case. Additionally, the process is more complicated due to the subtle shape of the skull mesh. In order to simplify our approximation while maintaining the skull mesh’s integrity simultaneously, we propose a solution that breaks down the entire approximation procedure into multiple subtasks by segmenting ROIs on the skull mesh and taking one of them for process each time. Processing each segmented ROI as an independent subtask represents a better performance in accuracy and simplicity during approximation compared to a direct process on the entire skull mesh.

The hierarchical procedure handles the regions that share dependency, such as lower jaw and chin. Considering this, we define the parent regions and their subregions (*Cranium* ↔ *Forehead*, *Jaw* ↔ *Chin*, *Orbit* ↔ *Eyebrow*, *Cheekbone*) as the first and second hierarchy, and the first hierarchical regions are deformed preferentially while the second hierarchical ones are handled. The ROIs and their hierarchies are shown in Figure 4.7.

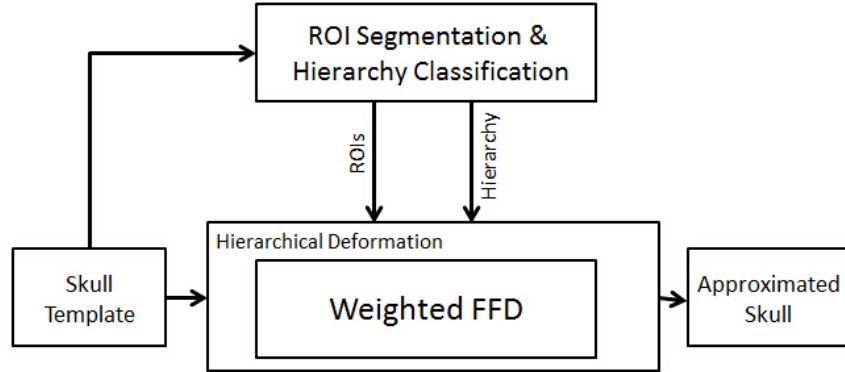


Figure 4.6: Adaptive approximation pipeline.

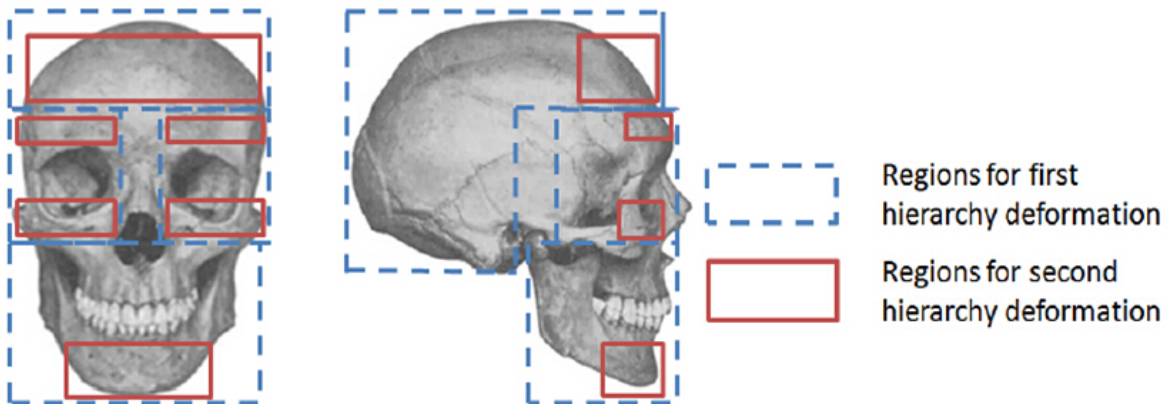


Figure 4.7: ROI segmentations and their hierarchy.

4.3.2 Weighted Free-Form Deformation

When trying to adapt a skull template mesh to an individual’s head mesh through shape modification, we propose a weighted FFD by adding two weight coefficients, weight of the control point and weight of ROI, in computation to obtain a precise deformation result. Control point weight, $w_{ControlPoint}$, defines a control point’s effect on the solid object during deformation; ROI weight, w_{ROI} , restricts the deformation effective only around the current ROI in the process. Our weighted deformation procedure is performed in four steps: *Parameterization*, *Control Points Generation*, *Control Points Manipulation*, and *Weighted Deformation Computation*.

Parameterization

Our deformation follows the routine in Sedberberg and Parry’s work [23] that encloses a solid object with a rectangular parallelepiped that is modeled with its own coordinate system represented by three orthogonal vectors as \mathbf{S} , \mathbf{T} , and \mathbf{U} . For convenience, we

describe the parallelepiped with a minimum bounding box of the object and align its local coordinate system \mathbf{S} , \mathbf{T} , \mathbf{U} with axes \mathbf{X} , \mathbf{Y} , and \mathbf{Z} in the Cartesian coordinates. Any arbitrary point $\mathbf{X}(x, y, z)$ on the object can be parameterized into local coordinates (s, t, u) as

$$X(x, y, z) = X_0(x_0, y_0, z_0) + s\mathbf{S} + t\mathbf{T} + u\mathbf{U} \quad (4.6)$$

where $\mathbf{X}_0(x_0, y_0, z_0)$ is the position of the local coordinates' origin in the Cartesian coordinates. Then the parameters s, t, u can be further interpreted as shown in Equation 4.7. For any point, its s, t, u values always satisfy the condition that $s, t, u \in (0, 1)$ because they are embedded in the minimum bounding box. The axis-aligned local \mathbf{S} , \mathbf{T} , \mathbf{U} coordinates system of the skull mesh is shown in Figure 4.8.

$$s = \frac{x - x_0}{S}, t = \frac{y - y_0}{T}, u = \frac{z - z_0}{U} \quad (4.7)$$

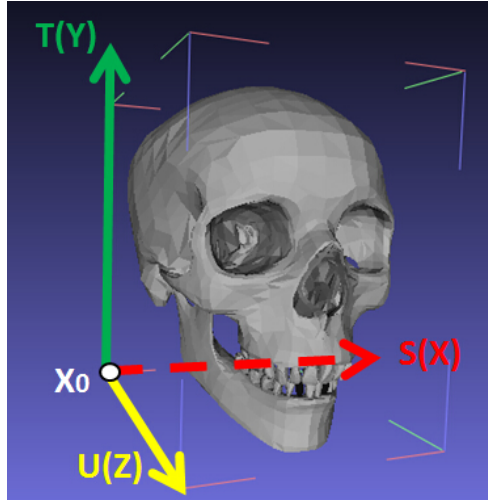


Figure 4.8: The \mathbf{STU} local coordinates of the skull mesh. Its \mathbf{S} is aligned with the \mathbf{X} -axis, \mathbf{T} with the \mathbf{Y} , and \mathbf{U} with the \mathbf{Z}

Control Points Generation

A group of $(3l + 1) \times (3m + 1) \times (3n + 1)$ control points are formed in a lattice structure with $l + 1$ planes in the \mathbf{S} direction, $m + 1$ planes in the \mathbf{T} , and $n + 1$ in the \mathbf{U} . Even though the control points lattice can theoretically be multiresolution with user-defined l, m and n , the resolution in our case is static with a preset layout ($l = 8, m = 12$, and $n = 8$) that was determined by aesthetic proportions. Each control point $\mathbf{P}_{ijk}(x_{ijk}, y_{ijk}, z_{ijk})$ is defined using Equation 4.8.

$$\begin{bmatrix} x_{ijk} \\ y_{ijk} \\ z_{ijk} \end{bmatrix} = \begin{bmatrix} x_0 + \frac{i}{l}S \\ y_0 + \frac{j}{m}T \\ z_0 + \frac{k}{n}U \end{bmatrix} \quad (4.8)$$

For every control point, we assign a positive weight w_{ijk} defining its influence. We prescribe a search radius on each control point from a circumscribed sphere generated depending on the size of the lattice cell. For a control point \mathbf{P}_{ijk} , if any points on the object is found inside that radius, we assign a weight w_{ijk} heavier than in a case when the detection result is null. Algorithm 1 gives a detailed description of the weight assignment of control points.

Algorithm 1 Control Points Weight Assignment

Input:

- The set of points on Head Mesh $\mathbf{X}(\mathbf{X}_1, \dots, \mathbf{X}_N)$;
- The set of control points: \mathbf{P}_{ijk} ;
- The local coordinate system: $\mathbf{S}, \mathbf{T}, \mathbf{U}$;
- Control point layout: l, m, n ;
- Variables: Control lattice's circumscribed sphere radius: R , the closest neighbour on the Head Mesh: $\mathbf{X}_{neighbor}$, distance between \mathbf{P}_{ijk} and $\mathbf{X}_{neighbor}$: d ;

Output:

- The set of assigned weight on each control point \mathbf{w}_{ijk} ;
 - 1: $R \leftarrow \text{computeCircumscribedSphereRadius}(l, m, n, \mathbf{S}, \mathbf{T}, \mathbf{U})$
 - 2: **for** $1 \leq i \leq l + 1, 1 \leq j \leq m + 1, 1 \leq k \leq n + 1$ **do**
 - 3: $\mathbf{X}_{neighbor} \leftarrow \text{searchClosestNeighbor}(\mathbf{X}, \mathbf{P}_{ijk})$
 - 4: $d \leftarrow \text{magnitude of } (\mathbf{X}_{neighbor} - \mathbf{P}_{ijk})$
 - 5: **if** $yes \leftarrow (d \leq R)$ **then**
 - 6: $w_{ijk} \leftarrow \text{assignNonEmptyWeight}(i, j, k)$
 - 7: **else**
 - 8: $w_{ijk} \leftarrow \text{assignEmptyWeight}(i, j, k)$
 - 9: **end if**
 - 10: **end for**
 - 11: **return** \mathbf{w}_{ijk} ;
-

Control Point Manipulation

The entire skull template mesh is segmented into several ROIs that they are processed one by one. In the control point manipulation, first we generate a bounding box for the specific region of deformation and enable the control points that are enclosed by or close to that box. Depending on the region's geometric shape, we determine the directions available for control point displacements. For each point $\mathbf{P}_{ROI}(x_{ROI}, y_{ROI}, z_{ROI})$ from the ROI, we search for the closest neighbour on head mesh $\mathbf{P}_{head}(x_{head}, y_{head}, z_{head})$ and compute their difference vector $\mathbf{V}_{difference}(x_{head} - x_{ROI}, y_{head} - y_{ROI}, z_{head} - z_{ROI})$. By analyzing all difference vectors and available displacement directions, we determine the resultant displacements on those selected control points. The manipulated control points \mathbf{P}'_{ijk} are later used in specifying the deformation computation.

Weighted Deformation Computation

Mathematically, the deformation function of FFD is a combination of the trivariate tensor product of Bernstein polynomial and coefficients of the displaced control points \mathbf{P}'_{ijk} from Step 3. For any point $\mathbf{X}(x, y, z)$ on the solid object for processing, its deformed position $\mathbf{X}'(x', y', z')$ can be determined in two steps: 1) parameterize its local (s, t, u) coordinates using Equation 4.7, and 2) compute its deformed position of x' , y' , and z' by evaluating the Equation 4.9.

$$\left\{ \begin{array}{l} x' = \sum_{i=0}^l \binom{l}{i} (1-s)(l-i) s^i \left\{ \sum_{j=0}^m \binom{m}{j} (1-t)(m-j) t^j \right. \\ \left. \left[\sum_{k=0}^n \binom{n}{k} (1-u)(n-k) u^k \times x_{ijk} \times w_{ijk} \right] \right\} \\ y' = \sum_{i=0}^l \binom{l}{i} (1-s)(l-i) s^i \left\{ \sum_{j=0}^m \binom{m}{j} (1-t)(m-j) t^j \right. \\ \left. \left[\sum_{k=0}^n \binom{n}{k} (1-u)(n-k) u^k \times y_{ijk} \times w_{ijk} \right] \right\} \\ z' = \sum_{i=0}^l \binom{l}{i} (1-s)(l-i) s^i \left\{ \sum_{j=0}^m \binom{m}{j} (1-t)(m-j) t^j \right. \\ \left. \left[\sum_{k=0}^n \binom{n}{k} (1-u)(n-k) u^k \times z_{ijk} \times w_{ijk} \right] \right\} \end{array} \right. \quad (4.9)$$

In order to obtain the skull's integrity globally, we assign a weight, here called weight of ROI or w_{ROI} , on every point and specify how much it is displaced in deformation depending on whether it belongs to the current ROI for processing. If the point is inside the ROI, w_{ROI} is set to be 100% and the deformation function is fully effective; otherwise, w_{ROI} is assigned with a percentage level from 10% to 50% based on its distance from the ROI. The closer it is from the ROI, the heavier percentage its w_{ROI} receives. Therefore, the weighted displacement vector $\mathbf{V}_{displacement}(\Delta x, \Delta y, \Delta z)$ of any point between its original position $\mathbf{X}(x, y, z)$ and the position evaluated from deformation function $\mathbf{X}'(x', y', z')$ is computed as $\Delta x = w_{ROI}(x' - x)$, $\Delta y = w_{ROI}(y' - y)$, and $\Delta z = w_{ROI}(z' - z)$. Finally, by applying the displacement vector $\mathbf{V}_{displacement}(\Delta x, \Delta y, \Delta z)$, we displace $\mathbf{X}(x, y, z)$ to its weighted deformed position $\mathbf{X}_{ffd}(x_{ffd}, y_{ffd}, z_{ffd})$ with Equation 4.10.

$$\begin{bmatrix} x_{ffd} \\ y_{ffd} \\ z_{ffd} \end{bmatrix} = \begin{bmatrix} x + \Delta x \\ y + \Delta y \\ z + \Delta z \end{bmatrix} \quad (4.10)$$

The complete weighted FFD procedure is summarized in Algorithm 2.

In comparison to the regular free-form deformation method, our weighted FFD method achieves a shape modification over ROI while the other parts maintain their shapes in integrity. With such a prerequisite, we are able to process the segmented regions on the

Algorithm 2 Weighted Free-Form Deformation of a Region of Interest

Input:

- The set of points on Head Mesh, $\mathbf{X}(\mathbf{X}_1, \dots, \mathbf{X}_N)$;
- The set of points on ROI, $\mathbf{P}'(\mathbf{P}'_1, \dots, \mathbf{P}'_M)$;
- The local coordinate system: $\mathbf{S}, \mathbf{T}, \mathbf{U}$;
- Variable: Control points layout: l, m, n , a set of control points: \mathbf{P}_{ijk} , $N \leftarrow$ size of \mathbf{X} ,
- Displacement Vector: $\mathbf{V}_{displacement}$, control points weight: w_{ijk} , ROI weight: w_{ROI} ;

Output:

- The set of points on deformed Head Mesh, $\mathbf{X}'(\mathbf{X}'_1, \dots, \mathbf{X}'_N)$;
 - 1: **for** $1 \leq i \leq l + 1, 1 \leq j \leq m + 1, 1 \leq k \leq n + 1$ **do**
 - 2: $\mathbf{P}_{ijk} \leftarrow generateControlPoints(i, j, k, l, m, n, \mathbf{S}, \mathbf{T}, \mathbf{U})$
 - 3: $w_{ijk} \leftarrow assignControlPointWeight(\mathbf{P}_{ijk}, \mathbf{X})$
 - 4: **end for**
 - 5: $\mathbf{V}_{displacement} \leftarrow manipulateDisplacementVector(\mathbf{X}, \mathbf{P}')$
 - 6: $\mathbf{P}_{ijk} \leftarrow displaceControlPoints(\mathbf{P}_{ijk}, \mathbf{V}_{displacement})$
 - 7: **for** $1 \leq I \leq N$ **do**
 - 8: **if** $yes \leftarrow insideROI(\mathbf{X}_i, \mathbf{P}')$ **then**
 - 9: $w \leftarrow 100\%$
 - 10: **else**
 - 11: $w \leftarrow assignWeights(\mathbf{X}_i, \mathbf{P}')$
 - 12: **end if**
 - 13: $\mathbf{X}'_I \leftarrow computeWeightedFFDFunction(\mathbf{X}_I, \mathbf{P}_{ijk}, w_{ijk})$
 - 14: **end for**
 - 15: **return** \mathbf{X}' ;
-

skull mesh hierarchically and perform a weighted free-form deformation of each region as an individual task. Since the entire skull mesh is decomposed into several regions and proceeded individually, the shape difference between the skull and head meshes on a particular region can be more precisely computed, and then, control points manipulated to specify a delicate geometric shape modification. When all regions are processed, the skull mesh is fully adapted as its shape is fitted to the user’s identity on the head mesh.

4.4 Stiffness Rendering

In order to model a physical-realistic deformation behaviour of a human head and reproduce such behaviour as stiffness to be integrated on the polygon mesh, our stiffness rendering procedure is processed in three main steps: (1) generation of a physical volume model, (2) modeling the displacement-pressure-relation with FEM and (3) integrating such behaviour as the stiffness property on the head polygon mesh.

4.4.1 Physical Volume Model

We tetrahedralize the merged polygon meshes from the user’s head and shape-adapted skull to obtain a volume mesh. In the volume mesh, the surface geometry of the head and skull is well preserved. In anatomy, a human head is composed of multiple materials, including skin, fat, muscle, and cortical bone and inside bone, forming into a layered structure. Since the given data is insufficient to determine the precise structure of user’s head, layers of skin, fat and muscle on the generated volume mesh are generalized as one, the *flesh* layer. Similarly, the layers of cortical and inside bone are generalized as another layer, the *bone* layer (see Figure 4.9). With a proper classification, the tetrahedral elements of the volume mesh can be categorized into two sets (layers) accordingly.

According to the anatomic structure, the *bone* set contains the elements of the core inside (or on) the boundary surface of the shape-adapted skull polygon mesh, and the *flesh* set includes those other elements between the boundary surfaces of the head and skull mesh. The classified layered structure of a head volume mesh is displayed in Figure 4.10. Referring to material data from [59, 60], the properties we assign to different elements are shown in Table 4.2. The volume mesh is further processed by constraining the vertex’s *Degree of Freedoms (DOFs)* belonging to specific facial regions, such as eyes, nose and lips, which are mainly supported by other organs than bone (skull).

Element Type	Flesh	Bone
Young’s Modulus E (N/m^2)	45658	6.0×10^9
Poisson’s Ratio ν	0.43	0.22
Density ρ (kg/m^3)	1060	1421

Table 4.2: Material properties of flesh and bone type elements.

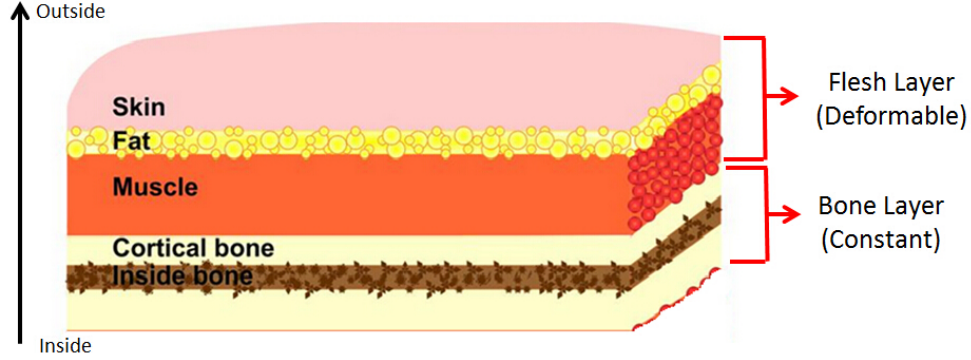


Figure 4.9: The flesh and bone layer of our volume model.

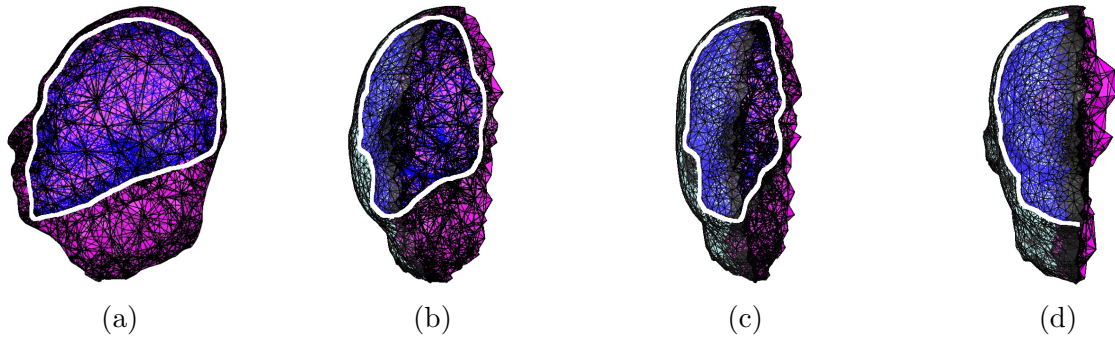


Figure 4.10: Cut view of the classified elements of a tetrahedral mesh. The boundary between *bone* and *flesh* sets is presented with a solid line. Elements inside belong to the *bone* set while those outside are classified as the *flesh* set.

4.4.2 FEM Simulation

Due to the continuum model's nonlinearity, the values inside the stiffness matrix change over time. Because of their variability, the stiffness matrix has to be re-validated dynamically in order to model the nonlinear deformation properly. Accordingly, the differential equations in a dynamic simulation can be written as

$$f_{ext} = f_{int}(u) + (M + K(u) + D)\dot{u} + M\ddot{u} \quad (4.11)$$

where f_{ext} is the force applied externally and $f_{int}(u)$ is the integrated stress that is caused by deformation u . \dot{u} and \ddot{u} are the first and second derivatives of displacement associated with time, here also referred to as velocity and acceleration, M is the stiffness matrix and D is the damping matrix. The resultant nodal displacement and velocity after each simulation time step can be computed by solving Equation 4.11. During integration, $f_{int}(u)$ is computed through the *Tangent Stiffness Matrix* $\mathbf{T} = \frac{\partial f_{int}(u)}{\partial u}$ for each element.

However, the computation of nonlinear dynamic simulation is highly expensive, that and very difficult to be achieved in real-time. Therefore, we initialize an integrator and start a dynamic FEM simulation by applying a continuous external force \mathbf{f}_{ext} on the deformable

object aligning with the normal. At the end of each time step, the resultant deformation is obtained from the integrator and used in static condition verification. If the condition is satisfied, the deformations are collected for analysis; otherwise, the simulation continues.

4.4.3 Head Polygon Mesh Stiffness Property Integration

From the FEM simulations, we obtain a collection of external force loads $f_{ext}(f_1, \dots, f_n)$ and the resultant deformations $u(u_1, \dots, u_n)$ where n is the collection's size as the deformation behaviour. With the discrete data, we model the behaviour in a continuous manner as the human facial soft tissue's stiffness property. For every interval between two adjacent data, its stiffness can be linearized with a simple form as $f = k_i \times u + b$ where k_i and b_i are two coefficients integrated into the polygon mesh as haptic modality. The values of k_i and b_i are computed using the deformation behaviour data as

$$\begin{cases} k_1 = \frac{f_1}{u_1} \\ k_2 = \frac{f_2 - f_1}{u_2 - u_1} \\ \vdots \\ k_n = \frac{f_n - f_{n-1}}{u_n - u_{n-1}} \end{cases} \quad (4.12)$$

$$\begin{cases} b_1 = 0 \\ b_2 = f_2 - k_2 u_1 \\ \vdots \\ b_n = f_n - k_n u_{n-1} \end{cases} \quad (4.13)$$

While a displacement u' on point \mathbf{P} is provided in a physical simulation, such as the intersection captured from a collision event, the proper coefficients k and b are selected accordingly to describe stiffness function as

$$\begin{cases} k = k_1, b = b_1 & (0 \leq d < d_1) \\ \vdots \\ k = k_n, b = b_n & (d_{n-1} \leq d < d_n) \end{cases}$$

Chapter 5

Results

5.1 Experiment Setup

In our experiments, we perform the proposed method to render stiffness property of 4 different subjects. The implementation of section 4.2 and 4.3 are realized using the *Point Cloud Library* [61], an open source project for processing 2D/3D images and point cloud data. The volumetric mesh (tetrahedral element type) is generated using TetGen [62], a tetrahedral mesh generation tool. The deformation simulation is processed by the VEGA FEM Library [63], a C/C++ physics library simulating deformation behaviour of 3D deformable objects. Due to the hyperelasticity of soft tissue material, we define our deformable model as the Saint-Venant Kirchhoff models [64] provided by VEGA. In a simulation, it only produces a pair of external force f_{ext} and its resultant deformation u that satisfy equilibrium conditions. As a result, for each point \mathbf{P} used in simulation, we assign a group of external forces (10 samples that divide a range from 0.5 N to 5.0 N equally, and such force range is selected because the haptic device's maximum force is 7.5 N) and apply one of them in an individual dynamic simulation procedure in order to gain sufficient paired information for deformation behaviour. Like most of the haptic rendering algorithms, the experimental external forces are aligned to the normal vector of the surface point in contact.

5.2 Landmark Detection and Models Registration Results

The corresponding facial features on head and skull points are detected and generated respectively. From the landmark detection, positions of a small group of facial features (eye corners, nose tip and nose corners) are localized on a 3D point cloud converted from the user's head polygon mesh (see Figure 5.1).

According to the detection results of our method on different users (5.1a-5.1d are of user A while 5.1e-5.1h are of user B), the features' position are correctly detected even when the variances in facial identity among subjects are obvious.

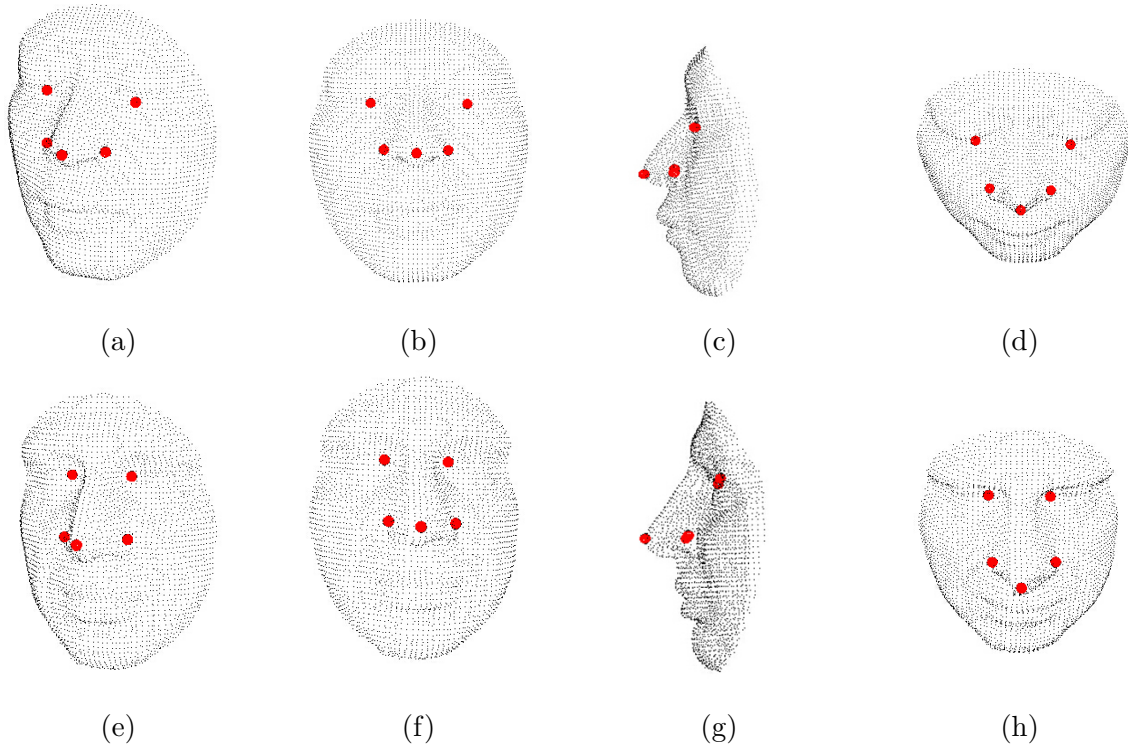


Figure 5.1: Detected facial landmarks on a 3D point cloud. The position of features are marked with a sphere. (a)-(d) are the detection result of user A from different views, and (e)-(h) are result of user B from different views. (a) and (e) are shown in isometric view. (b) and (f) are shown in frontal view. (c) and (g) are shown in profile view. (d) and (h) are shown in top view.

For the Models registration, we first obtain the corresponding facial features created on the template skull mesh according to aesthetic proportions, then get the semantically aligned skull and head point clouds from the *ICP* registration procedure. Figure 5.2 gives the result of our models registration of a user. The position of created features on the skull point cloud are shown in Figure 5.2b, and they approximately correspond with the detected features on the head point cloud.

From Figure 5.2c, we can see the skull point cloud is properly aligned to the head point cloud over a group of valid semantic correspondences. In general, the position of the registered skull point clouds in different experiments are close because of the similarities in human heads. However, the orientation of processed skull models varies slightly as it describes the user’s individual facial characteristics.

5.3 Skull Shape Adaption Results

Since our shape adaption approach is to segment the template skull mesh into several sub-regions and modify its shape using weighted FFD in a hierarchical order, for each specified

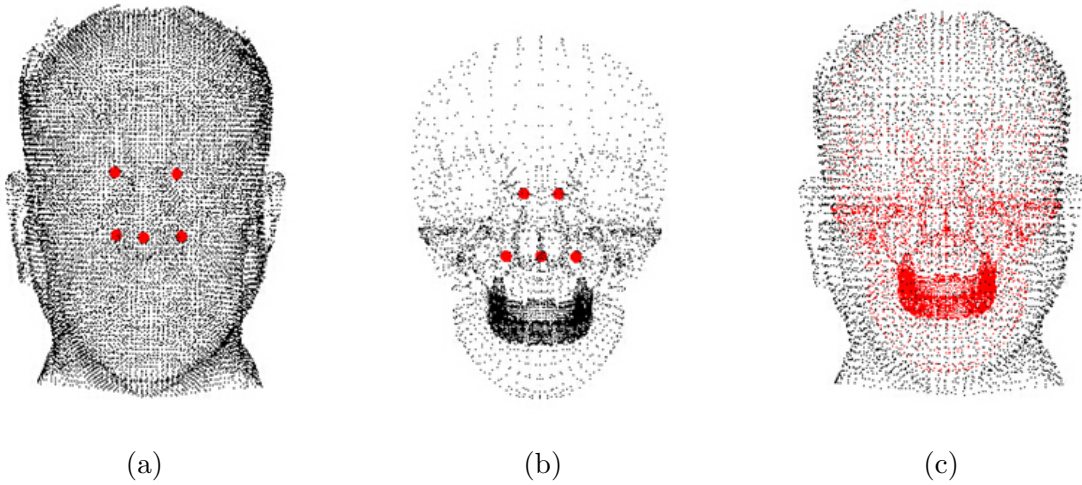


Figure 5.2: Results of models registration. (a) shows the facial features from detection results. (b) is the result of generated corresponding features on a skull point cloud. (c) displays the registered head and skull point cloud.

region we obtain a deformation result. Taking the Jaw, one of the segmented ROI, as an example, the deformation result (see Figure 5.3) reveals that its shape is significantly modified but other parts barely change. Such a result indicates the effectiveness of both weight variables, $w_{ControlPoint}$ and w_{ROI} , in our proposed method specifying the deformation and retaining object’s integrity as a whole.

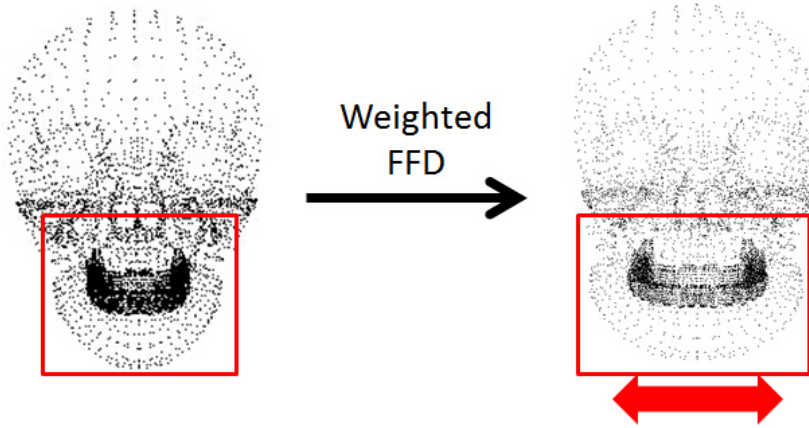


Figure 5.3: The local deformation result of the Jaw region obtained from the modified weighted Free-Form Deformation.

Figure 5.4 presents the fully processed skull models as the shape in all regions, modified to adapt different users’ head meshes. From the resultant approximation, the adapted skull approximately matches the head shape that shows the individual’s facial identity. Meanwhile, the integrity for the shape of different users is preserved to a satisfying level.

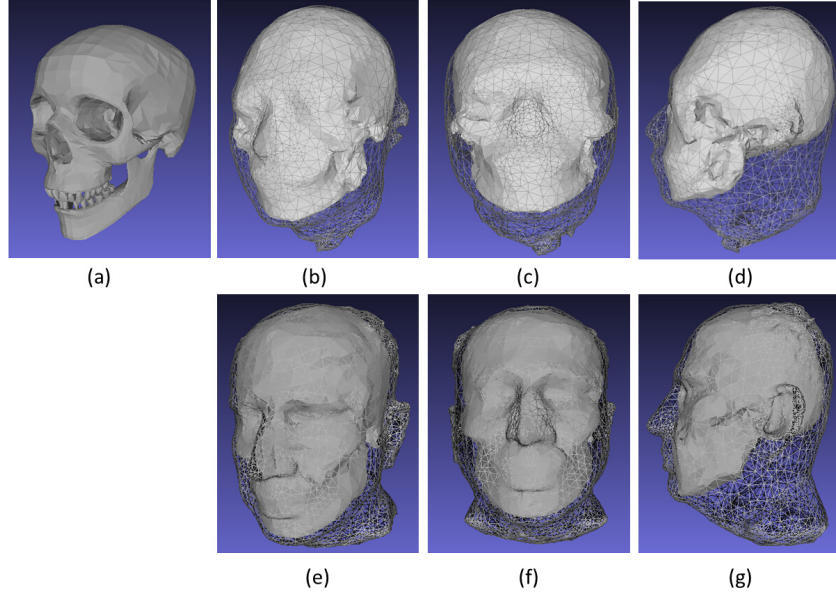


Figure 5.4: The fully adapted skull mesh as all ROIs are deformed. (a) is the original template skull mesh. (b)-(g) are the shape approximation results of two different users displayed in multiple views. (b)-(d) show the result of user *A*. (e)-(g) show the result of user *B*.

Therefore, it proves that our proposed *Hierarchical ROI* and *Weighted Free-Form Deformation* are feasible and are capable of producing a shape-adapted skull mesh from a given individual head polygon mesh.

5.4 Stiffness Rendering Results

From the FEM simulation, we receive a collection of the resultant displacements $\mathbf{u}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{10})$ under the effect of external forces $\mathbf{f}(\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{10})$ in equilibrium on the volume mesh's surface. For convenience, u and f are 3×1 vectors to real numbers as the vector's magnitude $u(u_1, u_2, \dots, u_{10})$ and $f(f_1, f_2, \dots, f_{10})$. The modeled deformation behaviour of a group sample features (Cranium, Forehead, Eyebrow, Cheekbone, Cheek, Chin, and Jaw) on the polygon mesh (see Figure 5.5a) are displayed in Figure 5.5b-5.5h respectively.

From observations of curves depicted from forces and deformations for each feature, we find a nonlinearity in the general behaviour as the deformations on the head model tend to stabilize at a level even when the external force strength is increasing.

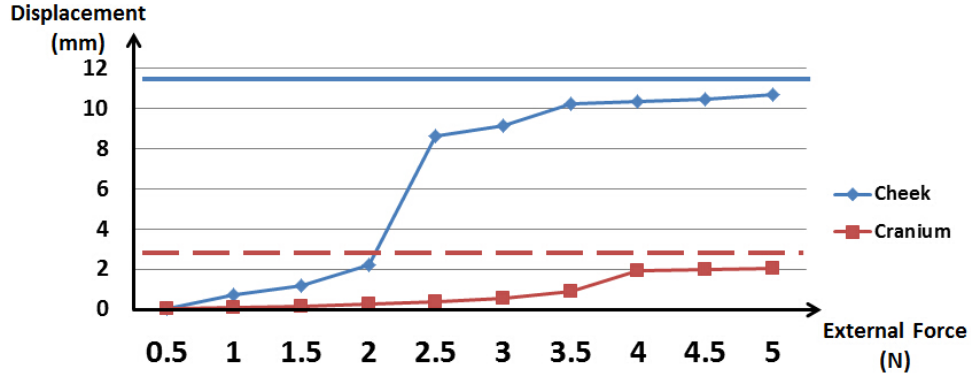


Figure 5.6: The simulation result of the features Cheek and Cranium are depicted with different markers respectively. The solid line is the distance between Cranium feature point and skull mesh, while the dash line shows the distance of Cheek feature.

Through a comparison of simulation results of different features, we observe variances in their deformation behaviour. First, for some features, their deformations vary obviously even when the same external force strength is applied. Additionally, the stabilized deformations on features are diverse: the Cheek has the largest deformation (10.67 mm), and deformation on the Cranium is the smallest (2.02 mm). Such diversity can be associated with the distance between the vertex for analysis on the head polygon mesh and the shape-adapted skull mesh beneath. For the Cheek and Cranium features as mentioned previously, their deformation behaviours and distances from skull mesh (Cheek’s distance is 11.38 mm while Cranium’s is 3.57 mm) are depicted in Figure 5.6. For the result evaluation, we assess the rendered stiffness by implementing a Human Head Haptic Rendering Application and inviting a group of human participants in a user study. The details of the implemented application and user study are discussed in Chapter 6.

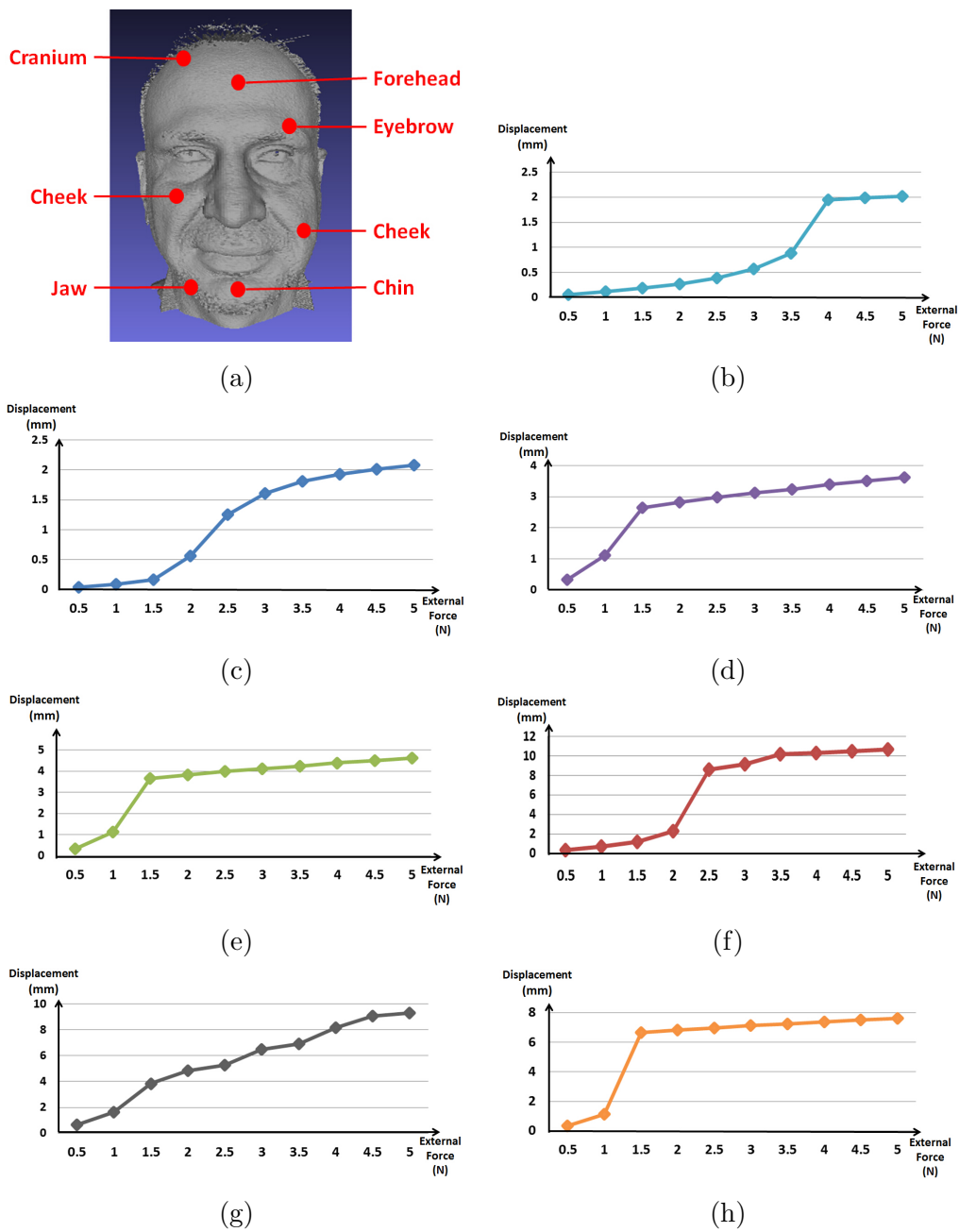


Figure 5.5: Deformation behaviour of each selected feature modeled from FEM simulation: the values on horizontal axis is the applied external force (N), and on the vertical axis is displacement (mm). (a) illustrates the features on a polygon mesh. (b)-(h) are modeled behaviours of features (Cranium, Forehead, Eyebrow, Cheekbone, Cheek, Chin, and Jaw)

Chapter 6

User Study of Rendered Stiffness

This chapter demonstrates the details about the implementation of a human head haptic rendering application that applies the rendered stiffness property from Chapter 4 to create a haptic interaction between a human user and a virtual head avatar.

6.1 Configuration

6.1.1 Hardware

The required hardware in our Human Head Haptic Rendering Application consists of a 3D head scanner to generate a human user's head polygon mesh, a PC to build the components in haptic rendering, a haptic device to display haptic feedback. The models of scanner and haptic device are shown in Figure 6.1 and the configuration of each specific hardware component is shown in Table 6.1.



Figure 6.1: Hardware models of 3D scanner and haptic device. (a) is the 3D head scanner²¹. (b) is the haptic device²².

²¹<http://cyberware.com/products/scanners/3030.html>

²²<http://www.novint.com/>

Hardware	Specifications
3D Scanner	<ul style="list-style-type: none"> • Model: Cyberware Model 3030/sRGB Color 3D Scanhead • Field of View: X or Theta: Depends upon motion platform; Y (at minimum Z): 300 mm (11.82 "); Y (at maximum Z): 340 mm (13.4 "); Z: 300 mm (11.82 ") • Sampling Pitch: X or Theta: Depends upon motion platform speed, 30 samples per second in X, typically 500 μm to 2 mm (0.02-0.08 "); Y: 350 μm (0.0125 "); Z (average): 75-300 μm (0.003-0.012 ") • Sampling Speed: 30,680 points per second, digitized to X, Y, Z and R, G, and B components • Color Data: 24-bit Red, Green, and Blue • Interface: USB 2.0
PC	<ul style="list-style-type: none"> • Model: Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz • CPU: Wireless-N • RAM: 8.00 GB • GPU: AMD Radeon HD 7600M
Haptic Device	<ul style="list-style-type: none"> • Model: Novint Falcon • 3D Workspace: 10.16 cm (4 in ") in each direction • Resolution: 400 dpi • Maximum Force: 8.9 N (2.0 lb) • Update Frequency: 1 kHz • Interface: USB 2.0

Table 6.1: Hardware Configuration.

6.1.2 Software

Multiple libraries and software programs are applied to implement haptic rendering in order to create a haptic interaction: 1) Virtual Studio 2013 is the C++ IDE (Integrated Development Environment); 2) OpenGL library visualizes the head avatar and haptic tool in the virtual environment; 3) Chai3D provides interfaces to connect the PC with the haptic device and generate force feedback on the device as shown in Table 6.2.

6.2 Deployment

The deployment of our Human Head Haptic Rendering Application is designed as in Figure 6.2. First, the user's head is scanned by a 3D scanner. A PC receives the scanning data and creates a 3D polygon mesh. At that PC, we integrate the rendered stiffness property obtained from Chapter 4 onto the polygon mesh as haptic modality, and implement other essential components for the haptic rendering application. The Haptic device displays the

Component	Software Configuration
PC	<ul style="list-style-type: none"> • Visual Studio 2013 • OpenGL 4.2 • Chai3D 2.0 • Windows 8.1

Table 6.2: Software Configuration.

haptic feedback between a human operator and a virtual environment, while the display device provides a graphical presentation of the virtual environment.

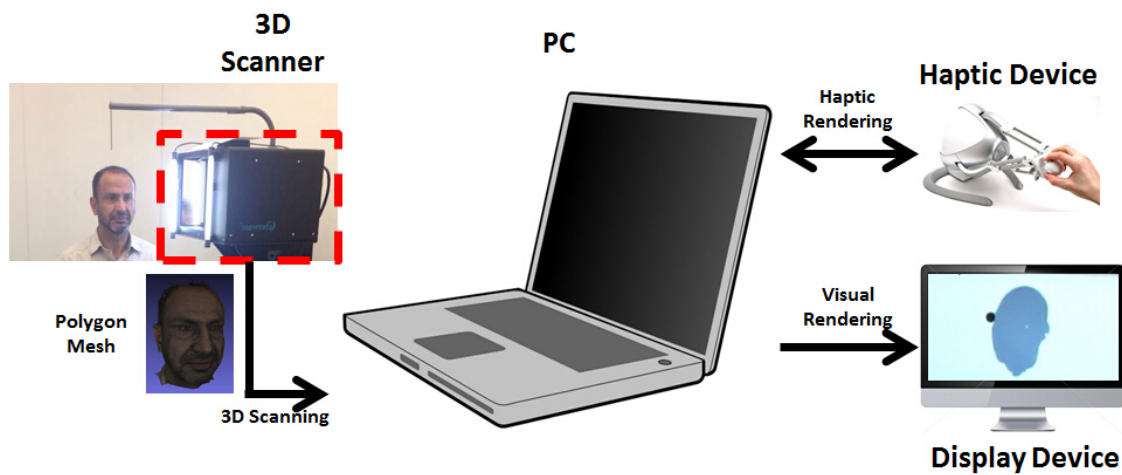


Figure 6.2: The deployment of the Human Head Haptic Rendering Application.

6.3 Overall System Architecture

Using the software libraries, I design and implement a Human Head Haptic Rendering Application, as illustrated in Figure 6.3. The application consists of several components including *Collision Detection*, *Coefficients Selection*, *Force Rendering*, *Haptic Update*, and *Graphic Update*.

- *Collision Detection* continuously detects any intersection between a haptic tool and head polygon mesh according to their geometric positions in the virtual environment. If a collision event is captured, the contact information (collision position X and intersection depth u) is recorded and then used in other computation procedures.
- *Coefficient Selection* chooses the proper stiffness coefficients from the polygon mesh's rendered stiffness property data according to contact information. Those selected coefficients are applied when computing interaction force.

- *Force Rendering* calculates the force value based on the specified coefficient, simulating the physical interaction between haptic tool and virtual head.
- *Haptic Update* initiates a bidirectional communication between human operator and virtual environment as it reads the position of the haptic device in the real world and generates a haptic feedback simulated from the virtual world.
- *Visual Update* renders interaction in the virtual environment through visualization.

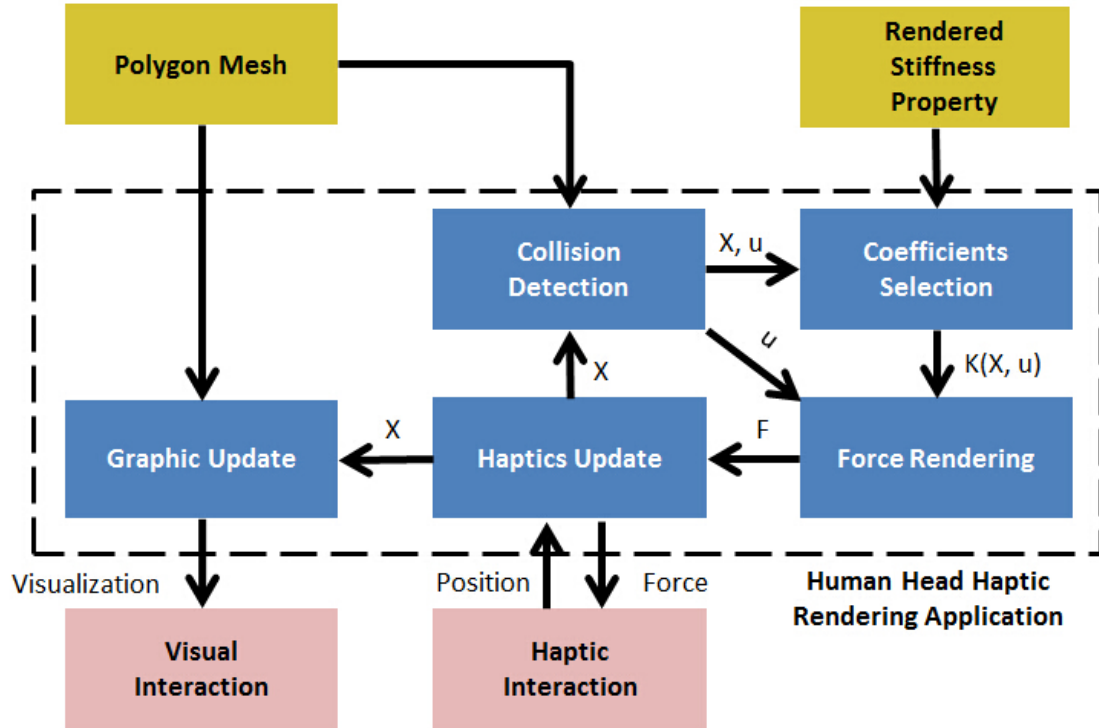


Figure 6.3: Overall system architecture of the Human Head Stiffness Rendering System.

6.4 Component Illustration

6.4.1 Collision Detection

Collision Detection is a vital component in our haptic rendering application as two other components, *Coefficients Selection* and *Force Rendering*, are dependent on the contact information (contact position and intersection) from an identified collision event. We employ a Bounding Volume Hierarchy (BVH) mechanism in collision detection for accuracy and efficiency. As a result, each facet, for both head and haptic tool objects, is wrapped in a bounding volume as a leaf node of a tree structure, while the entire mesh is enclosed in a single large bounding volume as the tree's root. A collision is identified when any volume

boxes from object and tool objects intersect. Furthermore, the positions of intersected volume boxes are used in computing the object-tool intersection. To further simplify the bounding volume construction, the Axis-Aligned Bounding Boxes (AABB) scheme is applied. Therefore, each volume box has its own max point $\mathbf{P}_{max}(x_{max}, y_{max}, z_{max})$ and $\mathbf{P}_{min}(x_{min}, y_{min}, z_{min})$. A collision event between the object box and tool box occurs only when values of $x_{max}, y_{max}, z_{max}$ from the head box are greater than $x_{min}, y_{min}, z_{min}$ from tool box, and $x_{min}, y_{min}, z_{min}$ from the head box are less than $x_{min}, y_{min}, z_{min}$ in the tool box.

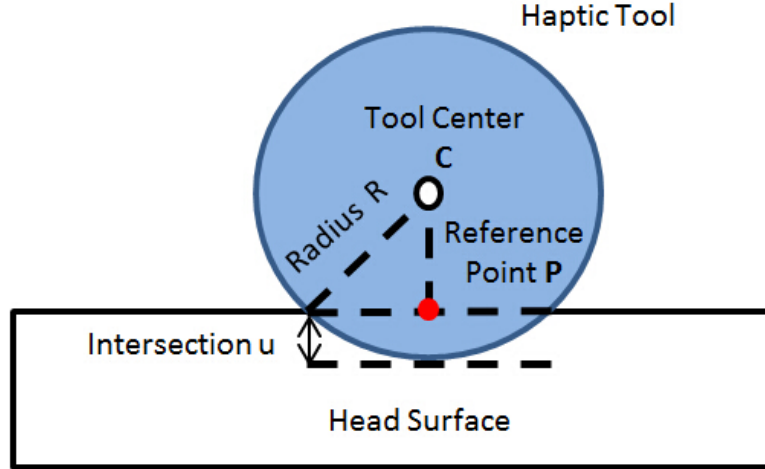


Figure 6.4: Intersection computation in a detected collision event.

When a tool-head collision is identified, the position of a tool object's center and the intersected points between the tool and head objects are selected for computing the contacted intersection. From those intersected points, the one which is closest from the tool's center $\mathbf{C}_{tool}(x_{tool}, y_{tool}, z_{tool})$ is considered as the reference point $\mathbf{P}_{reference}(x_{reference}, y_{reference}, z_{reference})$ indicates contact position \mathbf{X} . Since the radius of the haptic tool, R_{tool} is known, the tool-head intersection is described as $d = R_{tool} - |\mathbf{C}_{tool} - \mathbf{P}_{reference}|$ (see Figure 6.4).

From a detected collision, the contact information (contact position \mathbf{X} and intersection d) is passed to both coefficient selection and force rendering components for other procedures.

6.4.2 Coefficient Selection

The rendered stiffness property contains a large collection of coefficients that describes the stiffness levels reacting to different force systems' equilibrium conditions for each point on the head polygon mesh. With given contact information, we select the proper coefficients from the collection in order to reproduce the modeled deformation behaviour from stiffness rendering.

The selection is performed in two steps: 1) identify the polygon surface point depending on contact position \mathbf{X} , and 2) choose the exact coefficients based on intersection u . According to the results discussed in Chapter 5 where the modeled deformation behaviours vary significantly from feature to feature, the rendered stiffness coefficients are contact-position-dependent as well. With a specified contact position \mathbf{X} provided from a collision detection, we search the specific group of stiffness coefficients from the whole dataset collection. From the retrieved data group, $k(k_1, \dots, k_{10})$ and $b(b_1, \dots, b_{10})$, the exact coefficients are selected and returned to *Force Rendering*, the other value of contact information. The detailed selection method is shown as follows:

$$\begin{cases} k = k_1, b = b_1 & (0 \leq u < u_1) \\ \vdots \\ k = k_{10}, b = b_{10} & (u_9 \leq u) \end{cases}$$

6.4.3 Force Rendering

In *Force Rendering*, the force value resulting from tool-head interaction is computed and sent to Haptic Update for real-time haptic feedback. The force computation generates a simulated force value F from a linear equation as:

$$F = k \times u + b \quad (6.1)$$

where d is the tool-head intersection, and k and b are the selected stiffness coefficients in response to current contact. Since the coefficients k and b are parameterized from a modeled deformation behaviour of soft tissue, Force Rendering reproduces such behaviour displayed by force.

6.4.4 Haptic Update

Haptic Update is a component that associates the physical interaction between a human operator and the virtual world. Through a haptic device connecting with the computer, the procedures in this component simultaneously update the system *input* and *output*, the information exchanging from and toward the user respectively.

When updating *input*, the haptic device matches the tool's position manipulated by a user in the real world to the corresponding coordinate system in the created virtual environment. Since the device has 3 DOFs, the haptic tool's position is represented with a 3D vector containing the \mathbf{X} , \mathbf{Y} and \mathbf{Z} information in Cartesian coordinates as displayed in Figure 6.5. Since the orientation of Cartesian coordinates of the haptic tool and virtual scene are different, the tool's position has to be aligned to the corresponding position in the virtual scene. When a user is manipulating the haptic device, the haptic tool's position in both actual and virtual scenes are updated accordingly.

When a force value resulting from tool-object interaction is computed, the system updates the *output* by applying the recently computed force on the haptic device. Since the

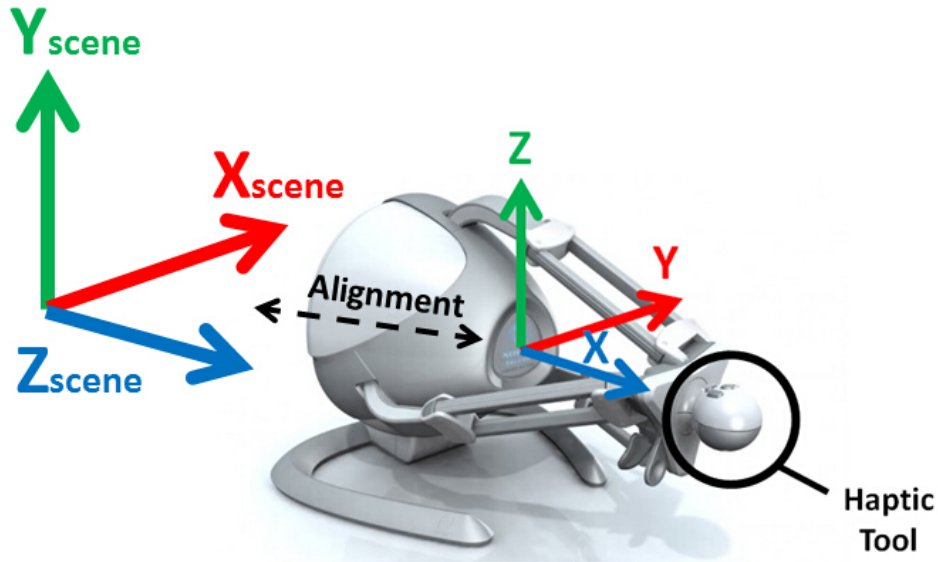


Figure 6.5: The haptic tool of a Novint Falcon device and its position represented in Cartesian coordinates.

virtual scene has three dimensions and the haptic device generates a 3-Degree-of-Freedom haptic feedback, the force-output cannot be described as a real number but as a 3D vector. We align the resultant force vector to the normal vector of the contact point on head polygon mesh and set its magnitude to be the value computed from force rendering. As the *Novint Falcon*, the haptic device we deploy for haptic feedback display, has a hardware limit where the maximum force is only 8.9 N , we constrain the updated force value to be 8.7 N when the computed result is larger than the limit to prevent the device from overheating.

6.4.5 Graphic Update

Graphic Update provides a visual presentation of the tool-head interaction in a virtual environment. In the virtual scene, a head polygon mesh is added and we represent the haptic tool with a sphere mesh by locating the sphere's center at the tool's position. Based on the aligned positions from the haptic update, we update the scene visually by transforming the sphere mesh to ensure its center is always located at the tool's position.

6.5 User Study

We performed a user study by inviting human participants to experience physical interaction on a stiffness-property-rendered 3D human head polygon mesh and evaluate each property depending on their given feedback. A group of features (Cranium, Forehead,

Eye, Cheekbone, and Jaw, sorted in descending order based on their positions on the y-direction) from 7 facial regions were selected because they were capable of describing most of the facial characteristics. The nose or eyes features were not selected because they were supported by other organs or structures rather than skull. The entire user study contains 4 procedures: *Tutorial*, *Physical-Interaction-Only Experiment*, *Physical-and-Visual Experiment*, and *Rationality Rating*.

6.5.1 Participants

25 participants (age: 27.25 ± 6.9 , 10 females and 15 males; 4 left-handed and 21 right-handed) participated in this user study. All participants signed a consent agreement. They were all students at the University of Ottawa, and their education levels included bachelor, master and Ph.D. Most of them (22 out of 25) were from Electrical Engineering, Computer Science and Computer Engineering, 2 were from Mechanical Engineering and 1 was from Social Science. None of them was reported to have any kind of somatosensory disorder. Based on their previous experience with haptic interface or devices, the participants were classified into two groups: Experienced, and Novice (7 Experienced and 18 Novice).

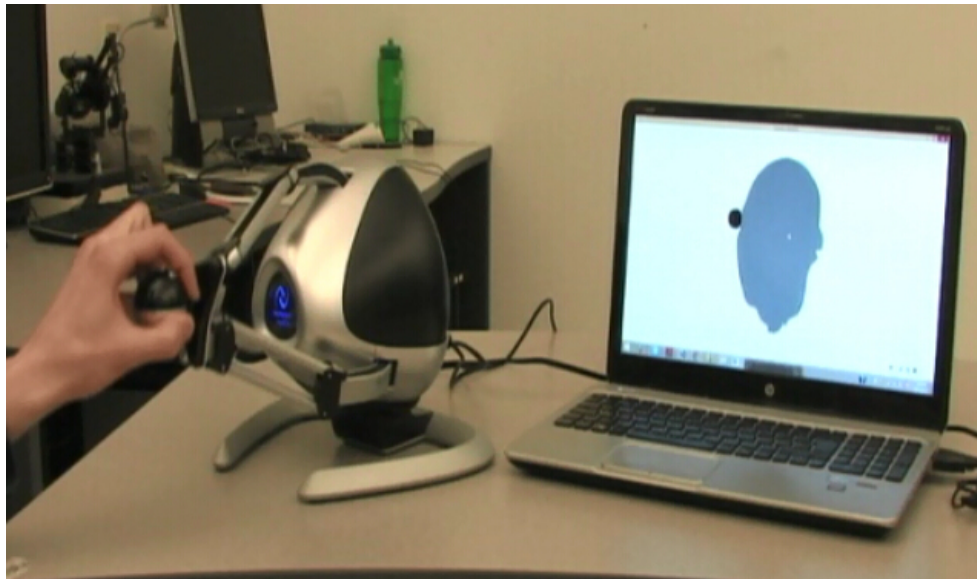


Figure 6.6: The Novint Falcon and other apparatus as configured in the user study. It shows a participant experiencing the rendered stiffness in a Physical-and-Visual Experiment. The screen is turned off temporarily while performing a Physical-Interaction-Only Experiment to make sure no graphic information is provided in that experiment.

6.5.2 Procedures

Tutorial

In the tutorial, the experimenter described and demonstrated the proper operation of the haptic device and participant could feel the stiffness property of each feature in a physical simulation. The appearance of the features for the experience was random. While all features were experienced, the regions with the toughest and softest stiffness level were provided to reconfirm if the participant could distinguish the stiffness difference over the head polygon mesh. At the end of the tutorial, the participant was asked if he/she could feel the different stiffness levels while the contact region changed. All participants gave positive feedback in telling the stiffness difference among regions.

Physical-Interaction-Only Experiment

In this section, we did not provide any graphical presentation and therefore the only interaction was the force feedback of the haptic device. The participant experienced the haptic feedback of the features on the head mesh according to their rendered stiffness. After the experience of each feature's haptic feedback, each participant's task was to give a rating from 1 to 10, where 1 was associated with the stiffness of the softest feature and 10 with the toughest feature. When all features were rated, there was another task for participants to select the two toughest and two softest features respectively from the previous experiment.

Physical-and-Visual Experiment

In contrast to Physical-Interaction-Only experiment, the graphical presentation in this section was available and was rendered with the physical simulation simultaneously. The task was the same as the one in 6.5.2, except with an extra task for participants to give a description of the features' contact positions based on visual feedback.

Rationality Rating

At the end of the evaluation, based on his/her expectation of the selected features' stiffness on an actual human head, the participant was asked to give a subjective rating of the rationality to see if the virtual head polygon mesh's stiffness property was rendered properly. The rating was chosen inside a range from 1 to 10 where 1 was totally unacceptable and 10 meant highly realistic.

6.5.3 Analysis

We obtained the ratings of both regions stiffness and rationality from participants' feedback. The statistical data is shown in Table 6.3 and 6.4 and details of the data analysis were extended in the following.

Regions	Participant Group					
	Experienced		Novice		Total	
	Mean	SD	Mean	SD	Mean	SD
Cranium	8.928	0.886	8.250	1.427	8.440	1.317
Forehead	7.928	1.367	8.000	1.485	7.980	1.426
Eyebrow	7.428	1.511	7.361	1.616	7.380	1.556
Cheekbone	5.571	1.592	5.389	1.065	5.440	1.202
Cheek	1.714	0.809	1.806	0.689	1.780	0.708
Chin	2.857	0.69	2.941	0.682	2.880	0.681
Jaw	5.571	2.09	4.806	1.742	5.020	1.834

Table 6.3: Ratings of region stiffness of overall and participant groups.

Rationality Ratings	Participant Group		
	Experienced	Novice	Total
7	2	2	4
8	2	4	6
9	3	8	11
10	0	4	4

Table 6.4: Rationality ratings of all participant groups.

Stiffness Region-Diversity Analysis

Ratings for each of the features are depicted in Figure 6.7, both the average and for each group of participants. From observation of the chart, the most overlapping curves indicated the ratings from participants were close, regardless of their groups. This phenomenon revealed that most participants agreed on the level of stiffness rendered on the head mesh. For each feature, the ratings from all groups were similar as the variance was small except for two (Cranium and Jaw) that were greater than 0.5. From the features rating, we observed both similarity and diversity. The similarity is that the rating of several features were close, and hence those features could be grouped accordingly. This may be caused by the similar integrated stiffness property. Diversity was presented as the significant difference between feature rating groups. As a consequence, the features were grouped, depending on their ratings, into 3 categories: Tough, Mediocre, and Soft, and the ratings inside a group were similar while were obviously different from group to group. Cranium (8.44, see Table 6.3), Forehead (7.98) and Eyebrow (7.38) were grouped as *High*, Cheek (1.78) and Chin (2.880) were classified as *Soft*, and Cheekbone (5.44) and Jaw (5.02) were taken as *Mediocre*.

To further evaluate the region-diverse stiffness property rendered on the head polygon mesh, we compared the features' ratings of both Physical-Interaction-Only and Physical-and-Visual experiments. Ideally, the ratings of a feature's stiffness, regardless of with or without graphical presentation, should be extremely close, as the prototype used in experiments was the same. In fact, since human tactility is highly subjective, the feature-

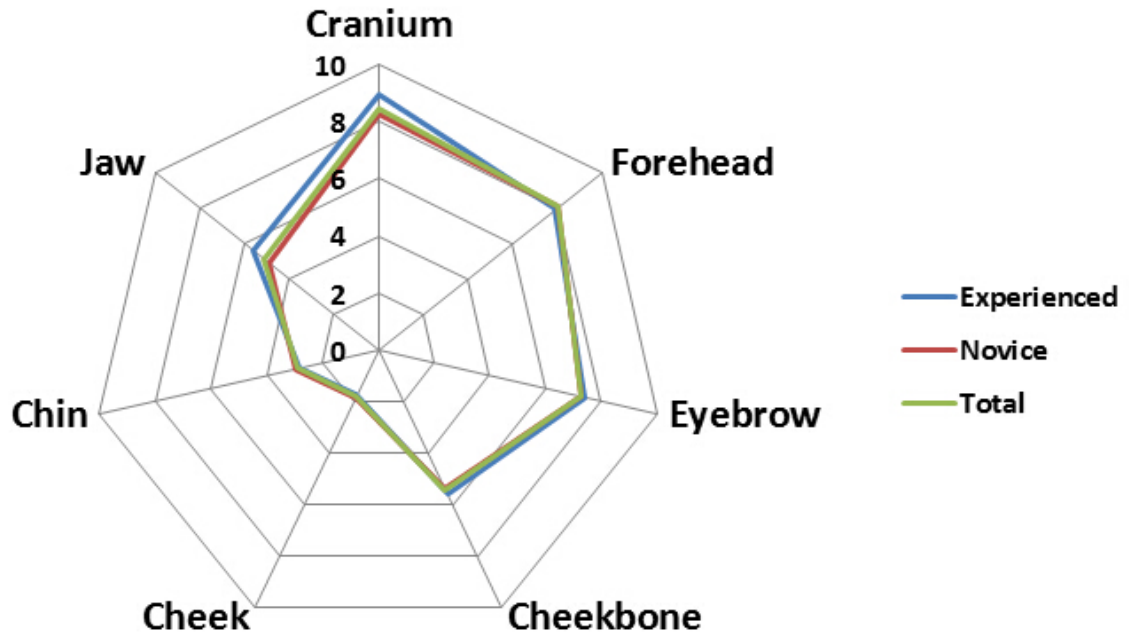


Figure 6.7: Average ratings of overall and for each participant group for each selected region.

stiffness ratings could be biased by the participants' expectations as they could see the feature in contact through visual perception. As a result, it was almost impossible to obtain an ideal comparison result. As shown in Figure 6.8, the stiffness of all features was rated higher (tougher) in the Physical-and-Visual experiment. The t-tests on 7 features, 3 of them (Jaw, Cheek, and Forehead) revealed significant differences with $p < 0.05$. Such differences are likely to be caused by participants' bias from graphical presentation. However, the similarity and diversity of ratings from previous observations were well retained in both objective (Physical-Interaction-only) and subjective (Physical-and-Visual) experiments.

By combining the analysis results from participant groups and both objective and subjective evaluations, we believe the contact position diverse stiffness property is rendered properly.

Stiffness Rationality Analysis

Even though the stiffness-rendered human head polygon mesh showed its region diversity in the previous analysis, its rationality was yet to be determined. Rationality represents whether the integrated stiffness of a virtual head polygon mesh can describe a physical property that satisfies people's expectation.

Empirically, the selected features for evaluation can be approximately ranked in a descending order depending on stiffness: Cranium, Forehead, Eyebrow, Cheekbone, Jaw,

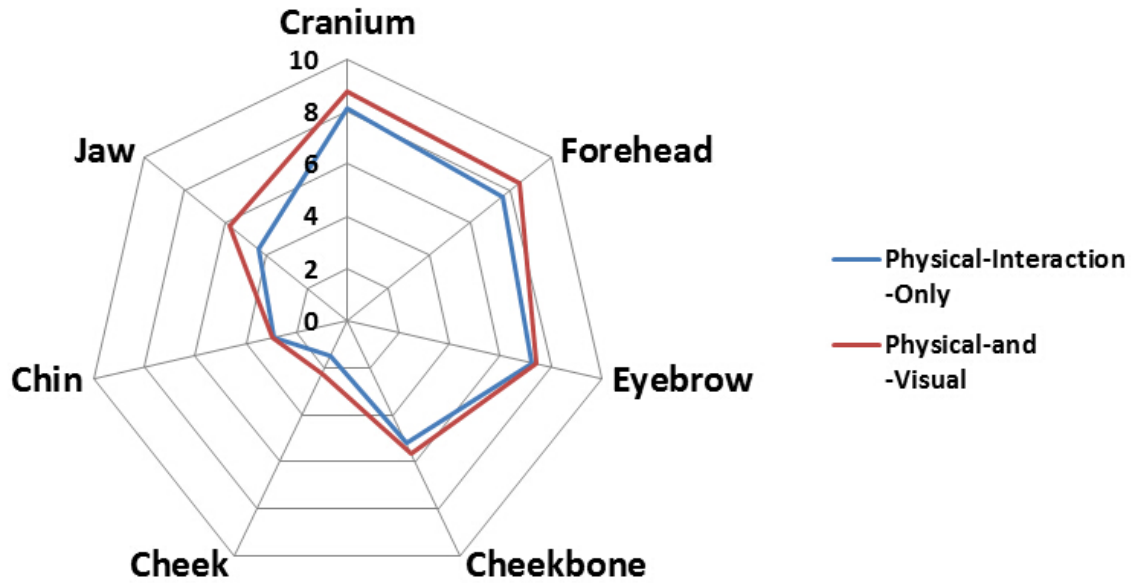


Figure 6.8: Average ratings of regions from the Physical-Interaction-Only Experiment and Physical-and-Visual Experiment.

Chin, and Cheek. In order to evaluate the rationality, we performed our analysis in two ways: 1) an accuracy analysis of tough/soft features selection, and 2) a distance analysis of participants' ratings and empirical rankings.

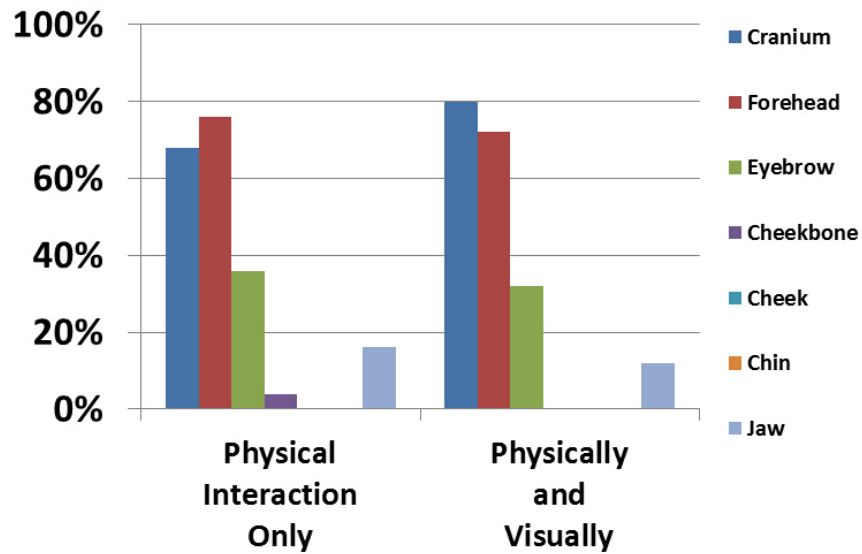
By comparing the accuracy from Physical-Interaction-Only and Physical-and-Visual experiments, the participants were more likely to provide valid results that match the empirical data with graphical presentation. In the soft features selection (see Figure 6.9b), where the Chin and Cheek are the two softest features, we observed an obvious accuracy enhancement, promoted from 72% to 92%, in the Physical-and-Visual experiment. Specifically, the accuracy for the Chin leaps from 64% to 88%. Even with existing exceptions like the correctness decrement for the Cheek and error increment for the Cheekbone, they could be considered as isolated because they did not affect the results in general. In the tough features selection (see Figure 6.9a) where the Cranium and Forehead are the two toughest features, the accuracy was slightly improved (from 72% to 76%) and selection of the Cranium, the feature with the toughest stiffness property, increased from 68% to 80%.

For the distance analysis of participants' ratings and empirical rankings, using each participant's stiffness ratings from the Physical-and-Visual experiment as they were more reasonable, we ranked them in descending order as R , and analyzed the distance in R and empirical rankings using Kendall's Tau [65]. In the comparison, we took every pair of elements i and j from empirical rankings and corresponding elements $R(i)$ and $R(j)$ from R , for matching analysis. If the elements satisfy such condition as $i > j$ and $R(i) < R(j)$, we call such pair as an *inversion*. Kendall's Tau counts total number of inversions

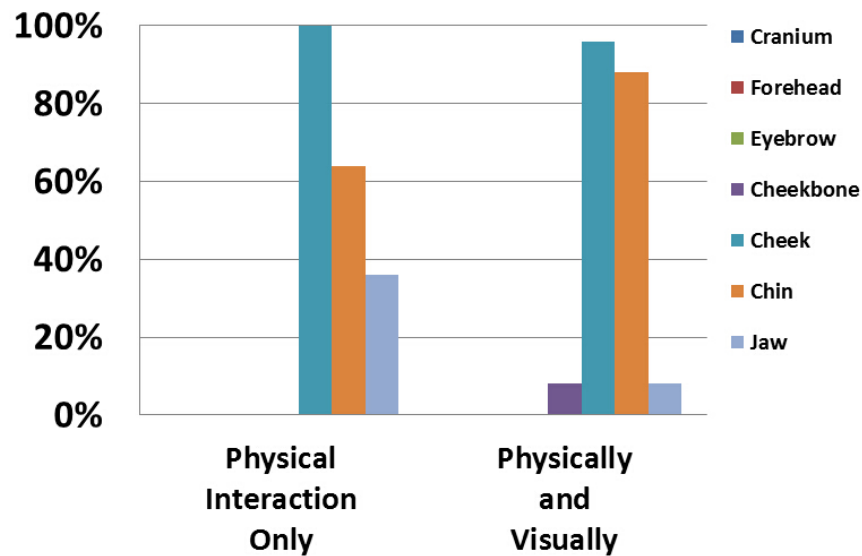
in R . The resultant accuracy was a ratio of correctly ranked pairs and total pairs as $\frac{TotalPairs - Inversions}{TotalPairs}$.

The resultant ranking accuracy of the entire participant group was reported as 91.86%, and 32% of participants (8 out of 25) had 100% accuracy in ranking, which means all facial features' stiffness were precisely ranked, while the others' accuracy varied from 66.7% to 95.3%. From the analysis of the results of tough/soft features selection and the stiffness rating ranking distance, we could objectively conclude that the polygon mesh's stiffness property was rendered properly.

In order to evaluate the stiffness's rationality subjectively, we analyzed the rationality ratings given by participants (see Figure 6.10). With a computation of rating means, the average of overall participants was 8.6 (S.D.=0.957), while the mean for experienced and novice user groups was 8.14 (S.D.=0.90) and 8.78 (S.D.=0.943) respectively. After a t-test on user groups' ratings, we did not see significant difference between the participants had haptic experience or not ($p > 0.05$). Looking at the ratings distribution (see Figure 6.10b and 6.10c), regardless of his/her haptic experience, most participants rated the rationality as 9. As expected, the experienced users were more critical while novice were more easily satisfied, as none from experienced group gave a highly satisfying rate as 10 but 28% of novice participants did. Additionally, 57% of the experienced group were lower satisfied that gave a rating less and group's average while the proportion of novice users was only 30%. In our evaluation, we performed a One Sample t-test of the given rationality ratings to assess if the rendered stiffness reached participants' expectations subjectively (here we determined the participant is satisfied with the rationality when his/her rationality rating is ≥ 8.0). The assessment result ($p = 0.025$) proved that participants agreed the stiffness was satisfying. As a result, the rendered property on the polygon head mesh is objectively and subjectively rational.



(a)



(b)

Figure 6.9: The distributions of participants' selection of tough and soft features. (a) is the distribution of the tough features selection. (b) is the distribution of the soft features selection.

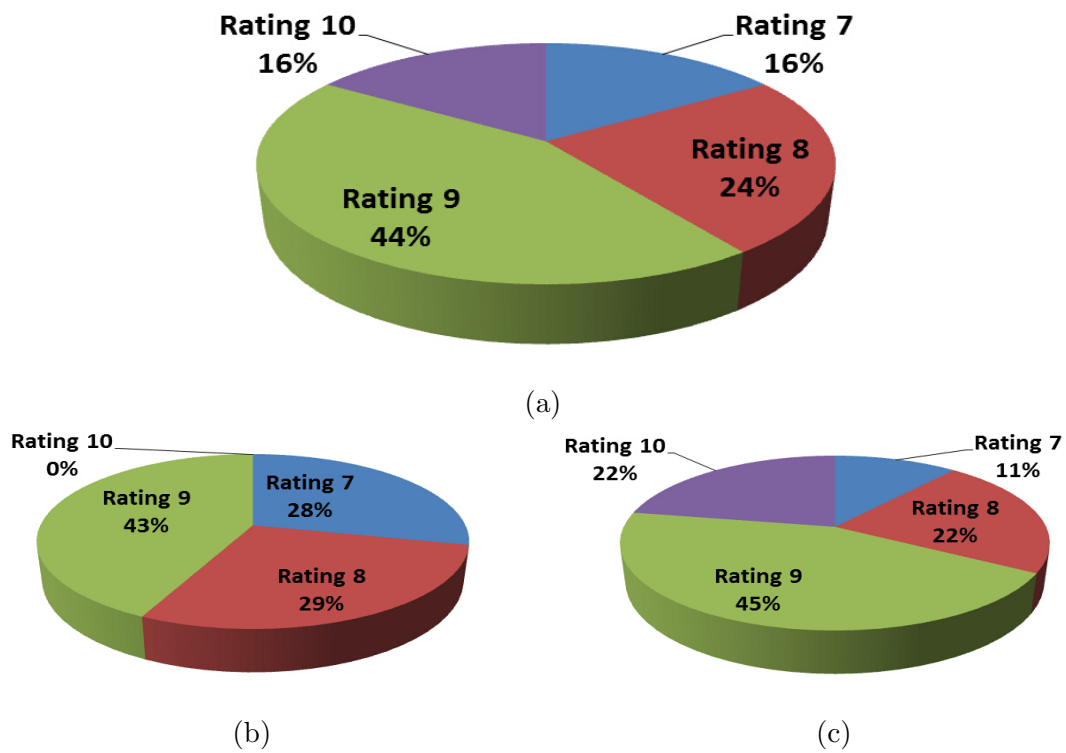


Figure 6.10: Rationality ratings of overall and for each participant group. (a) is the distribution of rationality ratings of overall participants. (b) is the distribution of the experienced group. (c) is the distribution of the novice group.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we presented a method to render the stiffness property of a human head polygon mesh. The rendered stiffness was defined by the human soft tissue material behaviour and represented as a collection of stiffness coefficients that were based on both contact region and deformation. In our approach, a template skull mesh was used as a reference in estimating head stiffness for its supporting function of human face structure. To model the soft tissue's material behaviour, our procedures included: 1) registering the template skull mesh to individual user's head polygon mesh semantically through a group of valid facial feature correspondences, 2) modifying shape of the registered skull mesh on several regions using hierarchical weighted Free-Form Deformation (FFD) to adapt to the user's face structure, and 3) modeling the soft tissue's material behaviour on a physical volume mesh generated from the head and shape-adapted skull mesh using continuum mechanics and mathematical framework as Finite Element Method (FEM) and rendering such behaviour into a collection of stiffness coefficients.

From the modeled material deformation behaviour of the head model, the stiffness property was determined to be region-dependent as the variance in equilibrium deformations resulting from the same external force load on facial regions were significant. Furthermore, the rendered stiffness revealed a nonlinearity as the equilibrium deformation tended to stabilize at a level even when the external force continued increasing. The stabilized level was related to the distance between a point on the head mesh and its closest neighbour on the shape-adapted skull mesh.

Based on the results of rendered stiffness coefficients, we built a Human Head Haptic Rendering System that integrated stiffness property on a scanned 3D head polygon mesh to create a physical interaction between the user and the touchable virtual head avatar. Additionally, we performed a user study by inviting human participants to experience the rendered stiffness of the head avatar. By analyzing the given feedback from participants, the stiffness property of the head avatar was determined to be properly rendered as it satisfied their expectation.

In summary, rendering the stiffness property of the human head polygon mesh greatly enhanced the haptic rendering to provide physical-realistic haptic feedback between a human operator and a touchable avatar. Such progress has significant potential to be applied in many fields, such as human-computer interaction, education and entertainment.

7.2 Future Work

There are several aspects of our stiffness rendering approach could be further developed for accuracy and efficiency in future versions.

- ***More delicate layered-structure of volume mesh*** The layered-structure of volume mesh in Chapter 4 is simplified and has only two layers: flesh and bone. If more detailed anatomy information is provided, we could build a more complicated structure by decomposing the volume mesh into more layers and assign them their own material property, such as skin, fat, muscle and bone. With a more precise physical continuum model, the human head's deformation would be modeled more physical-realistically.
- ***Powerful hardware or optimized algorithms to minimize computation time in processing nonlinear FEM.*** As discussed in Chapter 2 and 3, the computation of a nonlinear FEM model is highly costly and is difficult to be solved in real time because the stiffness matrix is not constant and has to be reevaluated at each time step. The large size of elements in the volume mesh makes it more challenging. If a powerful computer is used or the nonlinear FEM processing algorithms are optimized, the computation time would be significantly reduced and hence the stiffness could be rendered in real time.

APPENDICES

Appendix A

Human Head Mesh Stiffness Survey

Provided by *MCR Lab, Faculty of Engineering, University of Ottawa*

All information that you provided in this questionnaire is strictly confidential and thanks for your participation.

A.1 Overview

In this questionnaire, you are invited to participate in experiments designed and excuted by Minggao Wei, a researcher at MCR Lab, Faculty of Engineering, University of Ottawa, to evaluate the rendered-stiffness-property on a human head polygon mesh. The entire survey contains 3 parts: ***Background Information Collection, Tutorial*** and ***Experiments***.

- ***Background Information Collection***: some of your general information are collected and you may need to answer couples of questions related to our experiments. All the information you provide is strictly confidential.
- ***Tutorial***: you are about to learn how to manipulate the haptic device correctly following the instructions and demonstration of experimenter. Besides, you are expected to have a perceptual understanding of the stiffness properties rendered on the sample human head mesh when you complete the tutorial section.

Experiments: In the experiments, you need to participate in three experiments, Physical-Interaction-Only Experiment, Physical-and-Visual Experiment and Rationality Evaluation.

Physical-Interaction-Only Experiment: The only interaction in this experiment is the physical interaction displayed through a haptic device.

Physical-and-Visual Experiment: In this experiment, the interaction contains both vision with visual rendering the sample human head mesh and physical interactino displayed through a haptic device.

Rationality Evaluation: In this section, you are asked to evaluate the rationality of the rendered stiffness property of regions on human head mesh based on your experience in the former experiments.

I, _____, understand the procedures required in the experiments and would like to participate.

Signature: Date:

A.2 Participant Background Information

Gender: *Age:*

Educational Background: *Educational Level:*

Dominant Hand: a) Right b) Left

Somatosensory Disorder: a) Yes b) No

In which level do you have experience with haptic device: a) Experienced b) Novice

A.3 Tutorial

During the tutorial, vision interaction is not provided and you can only interact with the haptic device. The stiffness is based on the strength of force generated on the haptic device and the displacement of the haptic tool you are manipulating. There are stiffness properties of 7 regions you need to experience in this section. In the first part of tutorial, the 7 regions' appearance is selected randomly. For each region, you have 10-15 seconds to experience the stiffness. When we are moving to the next region, you will be informed. In the second part, the 7 regions' appearance is arranged in a descending order based on the stiffness. Also, you have 10-15 seconds to experience the stiffness of every region. At the end of the tutorial, you are asked a question to see if you are appropriate to participate the formal experiments.

Q1. Is it possible for you to distinguish the stiffness difference among regions in the tutorial by so far? a) Yes b) No

If the answer of *Q1* is *Yes*, you do not need to answer *Q2* as you are an appropriate participant for our experiments and we are ready to proceed the formal experiments. Otherwise, you have another opportunity to redo the tutorial section.

Q1. In the second trial in tutorial, is it possible for you to distinguish the stiffness difference among regions? a) Yes b) No

If the answer of *Q2* is *No*, you might not be an appropriate participant for our experiments. Thanks for your participation.

A.4 Experiments

A.4.1 Physical-Interaction-Only Experiment

In this experiment, you will be asked to experience the stiffness property on 7 regions on the sample human head mesh. The appearance of regions in experiment is not related to stiffness. Please give a rate of the stiffness level from 1 to 10 (1 is extremely soft, 10 is extremely tough). At the end of this experiment, you will be asked to select **2 regions** with the **toughest stiffness** and another **2 regions** with **softest stiffness** and write down their region index (from 1 to 7).

Region 1 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 2 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 3 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 4 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 5 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 6 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 7 Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Top 2 Toughest Regions:

Top 2 Softest Regions:

A.4.2 Physical-and-Visual Experiment

In this experiment, you will be asked to experience the stiffness property on 7 regions on the sample human head mesh. The appearance of regions in experiment is not related to stiffness. Firstly, please give a description of the contact position on the human head based on what you see. Then please give a rate of the stiffness level from 1 to 10 (1 is extremely soft, 10 is extremely tough). At the end of this experiment, you will be asked to select **2 regions** with the **toughest stiffness** and another **2 regions** with **softest stiffness** and write down their region index (from 1 to 7).

Region 1 Contact Position:

Stiffness Level: 1 2 3 4 5 6 7 8 9 10

Region 2 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Region 3 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Region 4 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Region 5 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Region 6 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Region 7 Contact Position: Stiffness Level: 1 2 3 4 5 6 7 8 9
10

Top 2 Toughest Regions:

Top 2 Softest Regions:

A.4.3 Rationality Evaluation

In this section you need to evaluate the rationality of the rendered stiffness property of regions rendered on human head mesh based on the experience in former experience.

Q3. Please give a rate of rationality level from 1 to 10 (1 is totally irrational, 10 is highly rational) of the rendered stiffness property of regions on the human head polygon mesh based on the comparison between the haptic feedback provided in the experiments and your experience and knowledge from normal life.

Rationality level: 1 2 3 4 5 6 7 8 9 10

A.4.4 Comments and Suggestions

Thanks for your participation. If you have any comments or suggestions, please write down here. Any comments or suggestions will be helpful in user study quality improvement.

References

- [1] K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: introductory concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, 2004.
- [2] El Saddik. The potential of haptics technologies. *IEEE Instrumentation Measurement Magazine*, 10(1):10–17, 2007.
- [3] Mandayam A. Srinivasan and Cagatay Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393 – 404, 1997. Haptic Displays in Virtual Environments and Computer Graphics in Korea.
- [4] A.E. Saddik, M. Orozco, M. Eid, and J. Cha. *Haptics Technologies: Bringing Touch to Multimedia*. Springer Berlin Heidelberg, 2011.
- [5] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles de-la Torre. Haptic interfaces and devices, 2004.
- [6] Mandayam A Srinivasan. What is haptics. *Laboratory for Human and Machine Haptics: The Touch Lab, Massachusetts Institute of Technology*, 1995.
- [7] Ming C. Lin, Miguel Otaduy, Ming C. Lin, and Miguel Otaduy. *Haptic Rendering: Foundations, Algorithms and Applications*. CRC Press., 2008.
- [8] C. Basdogan, S. De, J. Kim, Manivannan Muniyandi, H. Kim, and M.A. Srinivasan. Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64, 2004.
- [9] Alan Liu, Frank Tendick, Kevin Cleary, and Christoph Kaufmann. A survey of surgical simulation: Applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, 2003.
- [10] Wikipedia. Stiffness. <http://en.wikipedia.org/wiki/Stiffness>, 2015 (accessed May 29, 2015).
- [11] Multimedia Computing Reserach Laboratories (MCRLab). Touchable avatar. <http://www.mcrlab.net/research/touchable-avatar/>, 2015 (accessed May 29, 2015).
- [12] Wikipedia. Hooke’s law. [http://en.wikipedia.org/wiki/Hooke’s_law](http://en.wikipedia.org/wiki/Hooke's_law), 2015 (accessed May 29, 2015).

- [13] Assyr Abdulle and Bjorn Engquist. Finite element heterogeneous multiscale methods with near optimal computational complexity. *Multiscale Modeling & Simulation*, 6(4):1059–1084, 2007.
- [14] H. Dibeklioglu, A.A. Salah, and L. Akarun. 3D facial landmarking under expression, pose, and occlusion variations. In *Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2008.
- [15] Filareti Tsalakanidou, Sotiris Malassiotis, and Michael G Strintzis. Face localization and authentication using color and depth images. *IEEE Transactions on Image Processing*, 14(2):152–168, 2005.
- [16] F. Tsalakanidou, S. Malassiotis, and M.G. Strintzis. A 2D+3D face authentication system robust under pose and illumination variations. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 203–208, 2005.
- [17] Kyong I.Chang, K.W. Bowyer, and P.J. Flynn. Adaptive rigid multi-region selection for handling expression variation in 3d face recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 157–157, 2005.
- [18] K.I. Chang, W. Bowyer, and P.J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, 2006.
- [19] M.P. Segundo, C. Queirolo, O.R.P. Bellon, and L. Silva. Automatic 3D facial segmentation and landmark detection. In *Proceedings of International Conference on Image Analysis and Processing*, pages 431–436, 2007.
- [20] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn. A survey of rigid 3D pointcloud registration algorithms. In *Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, pages 8–13. IARA, 2014.
- [21] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [22] Rolf M Koch, Markus H Gross, Friedrich R Carls, Daniel F von Büren, George Fankhauser, and Yoav IH Parish. Simulating facial surgery using finite element models. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 421–428. ACM, 1996.
- [23] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH computer graphics*, volume 20, pages 151–160, 1986.
- [24] Josef Griessmair and Werner Purgathofer. Deformation of solids with trivariate b-splines. In *Proceedings of eurographics*, volume 89, pages 137–148, 1989.

- [25] Henry J Lamousin and Warren N Waggenspack Jr. Nurbs-based free-form deformations. *IEEE Computer Graphics and Applications*, 14(6):59–65, 1994.
- [26] Sabine Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, pages 187–196. ACM, 1990.
- [27] Ron MacCracken and Kenneth I Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 181–188. ACM, 1996.
- [28] Jules Bloomenthal. Medial-based vertex deformation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–151, 2002.
- [29] Jules Bloomenthal and Chek Lim. Skeletal methods of shape manipulation. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 44–. IEEE Computer Society, 1999.
- [30] Wenhao Song and Xunnian Yang. Free-form deformation with weighted t-spline. *The Visual Computer*, 21, 2005.
- [31] Dangxiao Wang, Youjiao Shi, Shuai Liu, Yuru Zhang, and Jing Xiao. Haptic simulation of organ deformation and hybrid contacts in dental operations. *IEEE Transactions on Haptics*, 7(1):48–60, 2014.
- [32] Seokhee Jeon, Seungmoon Choi, and M. Harders. Rendering virtual tumors in real tissue mock-ups using haptic augmented reality. *IEEE Transactions on Haptics*, 5(1):77–84, 2012.
- [33] Andrew Nealen, Matthias Mller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.
- [34] Gianni Borghesan, A. Macchelli, and Claudio Melchiorri. Interconnection and simulation issues in haptics. *IEEE Transactions on Haptics*, 3(4):266–279, 2010.
- [35] S. Payandeh and N. Azouz. Finite elements, mass-spring-damper systems and haptic rendering. In *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 224–229, 2001.
- [36] Mathieu Desbrun, Peter Schröder, and Alan Barr. Interactive animation of structured deformable objects. In *Graphics Interface*, volume 99, page 10. Citeseer, 1999.
- [37] Sergei Nikolaev. Non-linear mass-spring system for large soft tissue deformations modeling. *arXiv preprint arXiv:1403.2294*, abs/1403.2294, 2014.
- [38] Hing N. Ng and Richard L. Grimsdale. Computer graphics techniques for modeling cloth. *IEEE Computer Graphics and Applications*, 16(5):28–41, 1996.

- [39] N. Magnenat-Thalmann, F. Cordier, M. Keckeisen, S. Kimmerle, Reinhard Klein, and Jan Meseth. Simulation of clothes for real-time applications. In *Proceedings of Eurographics 2004, Tutorials 1: Simulation of Clothes for Real-time Applications*. INRIA and the Eurographics Association, 2004.
- [40] Yu Gao, H. Al Osman, and A. El Saddik. Mpeg-v based web haptic authoring tool. In *Proceedings of IEEE International Symposium on Haptic Audio Visual Environments and Games*, pages 87–91, 2013.
- [41] Karlsruhe Institute of Technology (KIT). Soft tissue mass-spring system model. <https://www.ibt.kit.edu/english/1984.php>, 2015 (accessed May 29, 2015).
- [42] L.P. Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. In *Proceedings of Computer Graphics International*, pages 156–165, 1998.
- [43] Shaoping Xu, X.P. Liu, Hua Zhang, and Linyan Hu. An improved realistic mass-spring model for surgery simulation. In *Proceedings of IEEE International Symposium on Haptic Audio-Visual Environments and Games*, pages 1–6, 2010.
- [44] U. Meier, O. Lopez, C. Monserrat, M.C. Juan, and M. Alcaiz. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, 77(3):183 – 197, 2005.
- [45] Matthias Müller and Markus Gross. Interactive virtual materials. In *Proceedings of Graphics Interface*, pages 239–246. Canadian Human-Computer Communications Society, 2004.
- [46] Yan Zhuang and John Canny. Real-time global deformations. In *Proceedings of 4th International Workshop on Algorithmic Foundations of Robotics*, volume 127, 2000.
- [47] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–36, 2001.
- [48] Xunlei Wu, Michael S. Downes, Tolga Goktekin, and Frank Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum*, 20(3):349–358, 2001.
- [49] Roger Temam and Alain Miranville. *Mathematical Modeling in Continuum Mechanics*. Cambridge University Press, second edition, 2005. Cambridge Books Online.
- [50] Wikipedia. Strain energy. <http://en.wikipedia.org/wiki/Stiffness>, 2015 (accessed May 29, 2015).
- [51] P. Boonvisut, R. Jackson, and M.C. Cavusoglu. Estimation of soft tissue mechanical parameters from robotic manipulation data. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4667–4674, 2012.

- [52] J. Teran, S. Blemker, V. Ng Thow Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 68–74. Eurographics Association, 2003.
- [53] Zhennan Yan, Lixu Gu, Pengfei Huang, Sizhe Lv, Xiao Yu, and Xianming Kong. Soft tissue deformation simulation in virtual surgery using nonlinear finite element method. In *Engineering in Medicine and Biology Society Engineering in Medicine and Biology Society*, pages 3642–3645, 2007.
- [54] Jianyun Chai, Jian Sun, and Zesheng Tang. Hybrid fem for deformation of soft tissues in surgery simulation. In *Proceedings of IEEE International Conference on Medical Imaging and Augmented Reality*, pages 298–303, 2001.
- [55] C.B. Zilles and J.K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 146–151 vol.3, 1995.
- [56] Thomas Harold Massie. *Initial haptic explorations with the phantom: Virtual touch through point interaction*. PhD thesis, Massachusetts Institute of Technology, 1996.
- [57] William R Mark, Scott C Randolph, Mark Finch, James M Van Verth, and Russell M Taylor II. Adding force feedback to graphics systems: Issues and solutions. In *Proceedings of 23rd annual conference on Computer graphics and interactive techniques*, pages 447–452. ACM, 1996.
- [58] P.J. Besl. *Surfaces in range image understanding*. Springer-Verlag, 1988.
- [59] Richard L Lieber. *Skeletal muscle structure, function, and plasticity*. Lippincott Williams and Wilkins, 2002.
- [60] Julie A Motherway, Peter Verschueren, Georges Van der Perre, Jos Vander Sloten, and Michael D Gilchrist. The mechanical properties of cranial bone: the effect of loading rate and cranial sampling position. *Journal of biomechanics*, 42(13):2129–2135, 2009.
- [61] Point Cloud Library. Point cloud library. <http://pointclouds.org/about/>, 2015 (accessed May 29, 2015).
- [62] Weierstrass Institute for Applied Analysis and Stochastics (WIAS). Tetgen. <http://wias-berlin.de/software/tetgen/>, 2015 (accessed May 29, 2015).
- [63] Jernej Barbič, Fun Shing Sin, and Daniel Schroeder. Vega FEM Library. <http://www.jernejbarbic.com/vega/>, 2012.
- [64] Jernej Barbic. *Real-time Reduced Large-deformation Models and Distributed Contact for Computer Graphics and Haptics*. PhD thesis, 2007.
- [65] D. Sengupta, U. Maulik, and S. Bandyopadhyay. Entropy steered kendall’s tau measure for a fair rank aggregation. In *Proceedings of National Conference on Emerging Trends and Applications in Computer Science*, pages 1–5, 2011.