**Doctoral Thesis**

# Modeling and Transfer of Inhomogeneous Haptic Data

Sunghoon Yim (임 성 훈)

Computer Science and Engineering

Pohang University of Science and Technology

2015

# 비균일 햅틱 정보의 모델링 및 전달

# Modeling and Transfer of Inhomogeneous Haptic Data

# Modeling and Transfer of Inhomogeneous Haptic Data

by

Sunghoon Yim

Department of Computer Science and Engineering
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A thesis submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy the Department of Computer Science and Engineering

Pohang, Korea
December 22, 2014
Approved by

SM Choi
Seungmoon Choi, Major Advisor

# 비균일 햅틱 정보의 모델링 및 전달

임성훈

위 논문은 포항공대 대학원 박사 학위 논문으로 학위 논문 심사 위원회를 통과하였음을 인정합니다.

2014 년 12 월 22 일

| | |
|---|---|
| 학위논문 심사위원회 위원장 | 최 승 문 (인) S.M. Choi |
| 위 원 | 안 희 갑 (인) |
| 위 원 | 정 완 균 (인) |
| 위 원 | 한 보 형 (인) |
| 위 원 | 한 성 호 (인) |

DCSE 20073594 임 성 훈, Sunghoon Yim, Modeling and Transfer of Inhomogeneous Haptic Data. 비균일 햅틱 정보의 모델링 및 전달, Department of Computer Science and Engineering, 2015, 107 P, Advisor: Seungmoon Choi. Text in English

# Abstract

Dealing with inhomogeneous haptic data is of importance in many haptic applications. In particular, biomedical data are essentially inhomogeneous, and medical haptic applications usually need sophisticated and efficient computational algorithms for modeling and simulating relevant signals generated from the interaction among inhomogeneous objects. This dissertation presents two core examples for modeling and transferring such inhomogeneous haptic data.

Our first effort focuses on capturing and rendering the behaviors of real objects of inhomogeneous deformation dynamics. We adapted a framework of the "data-driven haptics," where the response forces are modeled based on the recorded haptic data, and they are reproduced using an interpolation schemes. This approach can capture and display a diverse range of physical phenomena within one framework. In particular, inhomogeneity in lateral force, i.e., friction, in an unconstrained movement, e.g., stroking and rubbing for sliding exploration, can be accurately captured and rendered with our framework, which is one of the first attempts in the haptic modeling and rendering literature. The core point is the simulation of proxy point (actual contact point) movement based on sliding yield surface models, which possess necessary information for separating sliding and sticking states. In an off-line process, sliding yield surface models are built through an automatic palpation of a target real object. Internal radial-basis models are also modeled in the off-line process. During rendering, the movement of a proxy point is estimated using the sliding yield surface models, which becomes an input parameter of the radial basis interpolation models. Based on our performance evaluation, our framework shows less than 0.5N force error ratio in most cases, both in normal lateral direction.

The second example of this dissertation presents a computational algorithm for perceptually-correct haptization of inhomogeneous scientific data captured from the real environment. Inhomogeneity in scientific data may lead a user to misunderstand embedded information. An example of such cases is stiffness data captured by AFM, where conventional haptic rendering algorithms fail to deliver perceptually correct information to the users. In the worst case, a higher region with a smaller stiffness value can be perceived to be lower than a lower region with a larger stiffness value. This problem was explained by the theory of force constancy: the user tends to maintain an invariant contact force when s/he strokes the surface to perceive its topography. In this dissertation, we developed and evaluated a topography compensation algorithm for mesh object that can render the shape of a mesh surface and its stiffness distribution. This algorithm adaptively changes the surface topography on the basis of the force constancy theory to deliver adequate shape information to the user while preserving the stiffness perception. The performance of the proposed haptization algorithm is evaluated in comparison to the constraint-based algorithm by examining relevant proximal stimuli and carrying out a user experiment. Results demonstrated that our algorithm could improve the perceptual accuracy of shape and reduce the exploration time, thereby leading to more accurate and efficient haptization.

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

In general, haptic systems can be categorized by their goal. The first group tries to reproduce haptic responses of physical object, and the second focuses on an information delivery through a haptic modality. Since the two groups require different functionalities, each group needs a unique haptic model to store the haptic data as well as an appropriate haptic rendering algorithm to generate haptic signals. As our major thrust of this research is to deal with inhomogeneous haptic data, we select a representative but still challenging example from each group and develop two complete haptic systems that deal with inhomogeneous haptic data.

## 1.1 Reproducing Inhomogeneity

First system focuses on the reproduction of haptic responses generated during the interaction among physical objects of inhomogeneous material. Conventional haptic rendering algorithms could deal with variation in haptic properties, e.g., friction variation, but they could not fully handle property variations inside the object, e.g., inhomogeneity in stiffness beneath the object surface [102, 84]. Therefore, new rendering/modeling algorithms that can handle full inhomogeneity are necessary for many haptic applications. In particular, biomedical data are essentially inhomogeneous in the compliance and friction, and sophisticated algorithm to deal with those data are needed for many medical haptic applications, but there exists no systematic and general-purpose modeling/rendering algorithm so far.

In our first system, we developed a systematic, accurate, and efficient algorithm to capture, store, and render inhomogeneous stiffness and friction data. In our framework, we adapt "data-driven haptics" approach. This approach captures real signals resulted from physical interactions, stores them in an interpolation model, and use them to reproduce the captured signal. The advantage of this method is that this approach bypasses computationally complex contact dynamics simulation, so that the rendering algorithm become significantly efficient. This characteristic of the approach is very suitable for dealing with inhomogeneous data, since the analytic simulation of them is usually computationally complex.

In general, the core issue of data-driven haptics is how to select description set that accurately reflects the phenomena of real environments. In this dissertation, we made new description set that newly includes the position of simulated proxy position, in order to capture inhomogeneous stiffness and friction. In particular, in order to handle friction, previous data-driven haptic approaches are extended to support sliding exploration since sliding exploration is influenced by the contact and friction phenomenon of the deformable object. To this end, we use the contact based modeling and elasto-plastic analogy for describing the friction by introducing sliding yield surface models. Using sliding yield surface models, a proxy position can be simulated in real-time, which is used for the description.

In addition, we develop a shape modeling system based on contact modeling analysis for soft elastic object.

The final data-driven model could capture and display a diverse range of physical phenomena within one framework. In particular, this is among the first attempt that fully handle both stiffness and friction, which allows unconstraint interaction including sliding and stroking interactions.

## 1.2 Transferring Inhomogeneity

Our second focus is on the haptic perceptualization of inhomogeneous scientific data captured from the real environment. Inhomogeneity in scientific data may lead a user to misunderstand embedded information in case of an improper rendering algorithm for the information. For instance, for the case of transferring shape and stiffness data captured by AFM, the conventional haptic rendering algorithm for homogenous rigid object rendering may fail to deliver an exact topography information to the users. We present an algorithm (TCA-MESH) to haptize an object shape with a continuous stiffness distribution when the data are represented by a mesh. The goal of TCA-MESH is to deliver the shape and linear-elastic stiffness data represented in a mesh structure without perceptual distortion. To this end, TCA-MESH is designed to provide perceptually correct information when an appropriate EP is used, i.e., to render accurate topography during the user‘s stroking on the surface, while preserving correct stiffness perception during tapping. Stroking is the necessary EP for shape perception, whereas tapping is the optimal EP for stiffness perception [60]. Stiffness discrimination is also possible in case of stroking [60], but this is not pursued by TCA-MESH as a trade-off. In this aspect, TCA-MESH differs from general virtual object rendering that stresses the realistic simulation of physical phenomena.

## 1.3 Contributions

The contribution of the work can be summarized as follows:

- development of a data-driven haptic modeling and rendering method that supports sliding interaction with frictional responses
- improved haptization algorithm to transfer scientific data with inhomogeneity

## 1.4 Organization

In Chapter 2, previous works with respect to haptic modeling of the real object and haptization of the scientific data are represented. Chapter 3 describes the shape modeling method for real soft object. In Chapter 4, the data-driven modeling and rendering methods to model and transfer force response of the real object are introduced. In Chapter 5, we propose a haptization method to transfer the shape and inhomogeneous stiffness to the user. Finally, we conclude this study in Chapter 6.

Chapter 2

# Background and Related Works

## 2.1 Haptic Rendering

Haptic rendering can be defined as the process of generating contact force to create the illusion of touching virtual object [74]. Otaduy and Lin described the general definition of haptic renderig as follows:

*Given a configuration of the haptic device H, find a configuration of the virtual tool T that minimizes an object function $f(H-T)$, subject to environment constraint. Display to the user a force $\mathbf{F}(H,T)$ dependent on the configuration of the device and the tool.*

In this definition, we have one real system and one virtual system. The virtual tool acts as a virtual counterpart of the haptic device. This definition also assumes a causality precedence where the input variable is the configuration of the haptic device $H$ and the output variable is the force $\mathbf{F}$. This precedence is known as *impedance rendering*.

Then, the haptic rendering algorithms can be classified as direct rendering algorithm and virtual coupling. In the direct rendering algorithm, the tool configuration is directly assigned as the configuration of the haptic interface as $H = T$. Then, the force response can be determined $\mathbf{F}(T)$ directly. If we want deliver inhomogeneity of the environment, we can make inhomogeneous force response field directly. However, in order to enforce stability, we can separate the tool and device configuration , and insert in-between a visco-elastic link referred to as virtual coupling [20]. The connection of passive subsystems through virtual coupling leads to an overall stable system. In this method, the input of the rendering algorithm is the device configuration, bet the tool configuration is solved in general through an optimization problem, which also accounts for environment constraints. The difference between the tool and configuration is used both for the optimization problem and for computing output device forces. The most common form of virtual coupling is a visco-elastic spring damper link. Such a virtual coupling was used in god-object [102] and virtual proxy [84] algorithms for rendering rigid object. The elastic and viscous coefficients of the spring damper link are selected to maintain stable system. These parameters cannot directly used to represent the compliance and viscosity of the target environment.

In order to render the elastic properties inside the object, voxel based rendering [2] or deformable object haptic rendering using a variation of finite element methods (FEMs). The voxel based rendering to display the volumetric data will be introduced in section 2.3 The rendering methods for transferring data are similar to the data visualization techniques in graphics fields. This kind of haptic rendering algorithms are often called the data haptization algorithm or data perceptualization algorithm.

In order to render haptic realistic interaction with soft object for surgical simulation, FEM is frequently used. FEM is a numerical analysis technique for obtaining approximate solutions to a wide variety of engineering problem. The finite element discretization procedures reduce the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in term of assumed approximating functions within each element [41]. It is the most common approach to solve problems based on continuum mechanics Numerical problems of the contact mechanics problems, such as the tool contacting with the soft objects, were solved using FEM [52, 98]. Using these techniques, we can use standard material parameters like Young‘s modulus and Poission ratio, and so on. However, the drawback of FEM based algorithms is computational cost, thus, most of real-time haptic rendering algorithms using a variation of FEM are used pre-computation [22, 45, 9, 3]. The deformation object rendering with frictional contact was also developed [27].

For the haptic rendering of the elastic or visco-elastic material without considering the global deformation, reality based modeling is also proposed and developed. The detailed introduction of the reality based modeling will be represented as follow.

## 2.2 Haptic Modeling of Real Objects

Haptics research has seen persistent effort put into haptic modeling. In general, the haptic modeling can be classified into two approaches: synthetic modeling and reality-based modeling. Synthetic modeling relies on manual authoring by experts using dedicated software tools. It has been applied to modeling of object shape and other haptic properties such as stiffness, friction, and texture [63]. Those for tactile stimuli have also been active [91, 86, 61]. This approach can lead to higher quality models, but with increased cost.

In this dissertation, we focus on the reality-based on modeling. Reality-based modeling captures the haptic properties of real objects by measurements. This technique creates the virtual environments based on real-world interaction. The data recorded during contact between an instrumented tool and a real environment are used to generate reality-based models. The result model can be a database of recorded responses to various haptic stimuli, an empirical input-output mapping, or a set of physics-based equations [73]. In data-driven approach, recordings of a movement variable, such as position or force, are played back during haptic rendering. Input-output models are created by fitting or interpolating the recorded data with no or only little prior knowledge about the material characteristics. Physics-based models are constructed from a fundamental understanding of the mechanical principles underlying the recorded haptic interaction.

### 2.2.1 Data Acquisition Methods

Many objects parameters including shape, stiffness, friction and texture can be recorded for haptic recreation in virtual environments. For the shape modeling, the optical 3D scanner is used popular. This standard method in computer graphics can sample a huge number of surface points rapidly and precisely, based on the reflection or scattering of light on the surface of a real object [6]. However, the optical scanner cannot be appropriate for objects of certain optical properties (e.g., transparent). Contact-type modeling devices using a mechanical probe also used for shape modeling. For example, the commercial MicroScribe of GoMea- sured 3D is a robotic 3D digitizer for shape modeling. Using the MicroScribe, the user scans the surface of a rigid object, and these trajectories are converted to a geometric model of the object. This kind of tools called as coordinate-measuring machine.

Acquisition of haptic data is inherently required a contact. The contact probe can be controlled by the computer [77, 32, 56] or be explored by the user [40, 57, 89]. The various sensors are installed to the contact probe to get the required data. Pai et al. proposed a robotic measurement facility (ACME) consisting of a contact probe with a force sensor, a commercial 3D stereo-vision camera, and a standard RGB camera [77]. Computed-controlled robot arms position the contact probe and the cameras to measure the shape, deformation, and texture of an object. The same robotic measurement facility was also used to estimate elastic deformable models of 3D objects in [56]. Lang and Andrews uses the portable texture scanner of [76] and two more cameras for modeling of shape, texture, and stiffness [57]. Romano et al. modeled and synthesized textures only using an accelerometer [83]. Fong also introduced a 3D haptic data acquisition system, which makes use of a PHANToM device and optical scanner [32]. He employed dense point sampling strategies to capture the haptic information. The framework of Hover et al. can capture real visco-elastic responses of homogeneous object, and re-render the modeled responses, all using the same haptic interface with a force sensor [40]. Sianov and Harders improved Hoever's framework to capture and render the inhomogeneous object [89]. They adopted two steps data acquisition: a broad phase object sampling and a detailed acquisition at representative locations to handle inhomogeneity.

### 2.2.2 Modeling Approaches

For physics based model, parameterized behavior models are extracted from the measured force signals. Maclean developed the Haptic Camera [68]. In this work, parameters like stiffness, damping, and inertia were obtained by fitting a piecewise linear model to recorded force signals. For surgical cutting applications, cutting forces from surgical scissors were acquired and used for haptic feedback by tuning the piecewise linear models [35, 72]. In a similar manner, Edmunds and Pai modeled the recorded force signals from the bone-pin placement task by a polynomial function [30]. Colton and Hollerbach modeled force and acceleration signals from push-buttons by fitting a non-linear model [21]. Kry and Pai modeled the interaction between the human hand and and rigid objects by capturing the motion data and the contact force to estimate the compliance of the finger joints [55]. For the compliance model, Lang et al. employed a linear elastic continuum mechanics model represented as a discrete Green's functions matrix to describe the captured data by measurement [56]. Schoner et al. improved Lang et al.'s framework by capturing dynamic behavior with a spring-damper model [88]. Judd et al. identified stiffness, damping, and inertia model using least-squares approximation using a commercial haptic interface with a force sensor in [50]. Yamamoto et al. [100] estimated stiffness maps for material using the non-linear Hunt-Crossley model [42]. Jeon and Choi [48] extracted the parameters for the Hunt-Crossley model and the modified Dahl friction model [24] to describe the behaviors of homogeneous real object to achieve stiffness modulation in haptic augmented reality environment. These studies required the physical model. However, the model may include simplified assumption about the target meterial.

In a model-free data-driven rendering strategy, multi-quadratic RBFs were used for the approximation 3-DOF elastic force responses in [32]. In that system, single RBF model was trained for one contact point. For one contact point, the acquisition was limited to push and release operation. The result force was the interpolation of the several RBFs for the sampled contact point. Hovers et al. also [40, 39] employed RBFs to reconstruct a variety of material properties such as visco-elasticity and purely viscous behavior. In that framework, the interaction was constrained to one contact point and assumed a homogeneous object. Moreover, the reconstruction algorithm could not accommodate higher dimensions and sparse sampling Then, Sianov and Harders [89] were proposed the data-driven method to capture inhomogeneity and to accommodate higher dimension information. However, this approach could not hand the sliding exploration. The system only can capture the force response in the stick state. Thus, in order to archive unconstrained data acquisition and restoration, the sliding exploration must be considered.

## 2.3 Haptic Transfer of Data

In *data perceptualization*, the properties of a dataset are conveyed to the user through multimodal sensory channels including vision (visualization), sound (sonification), and touch (haptization). Data perceptualization enables the user to see, hear, and/or touch the data with an increased bandwidth of information transmission [8]. One of the components central to effective data perceptualization is a *transfer function*, which maps a data variable (e.g., the density of a voxel) to a display attribute (e.g., color, pitch, or force). A transfer function must guarantee that the information *perceived* by the user matches the original information contained in the dataset. Otherwise, perceptualization can give the user an incorrect understanding of the data properties.

The work reported in this dissertation is concerned with data *haptization* involving the sense of touch. A pioneering work of data haptization was the GROPE project by Brooks et al. [10], which enabled the user to perceive and control the docking of molecules in a large virtual environment using a large stereo display and a force-feedback haptic interface. This was followed by Taylor et al., who developed a teleoperation system that interfaced a force-feedback device with a scanning tunneling microscope to present both a visual and a haptic display of nano-scale data to the user [94].

Then, the research interest shifted to the haptization of volumetric data. Early algorithms for haptic volume rendering used the local gradients of voxels to determine a force transfer function [44, 2]. The traditional proxy-based haptic surface rendering algorithms were also extended to deal with volumetric data [66, 43]. For scientific data perceptualization, Lawrence et al. proposed several haptization methods for scalar, vector, and tensor fields [58, 59]. They used intermediate representations, e.g., stream lines, to render the different properties of vector and tensor fields. Later, Lundin and his colleagues published a series of papers on proxy-based haptic volume rendering. They began their research by using haptic primitives, such as points, lines, and surfaces, to build abstraction layers for volumetric rendering [65]. This work was improved by using haptic modes, e.g., force, follow, and surface modes, as the building blocks for constructing a haptic interaction scheme appropriate for a dataset [64, 78]. Each haptic mode is designed by selecting multiple haptic primitives that are used in this particular mode. Palmerius and Forsell also evaluated the effects of haptic modes on the user's ability of identifying structures within volumetric data [80]. In addition, Maciejewski et al. presented a volume rendering algorithm to simulate the interaction between multiple volumetric datasets, with an application to molecular docking [67]. For a further comprehensive review of general data haptization, see a recent survey paper of Panëels and Robert [81].

Another topic that has received increased attention pertains to *simultaneous haptization of multiple haptic attributes*. This research has focused on the rendering of surface shape and stiffness, the two most fundamental properties that need to be provided in the majority of haptics applications. Although virtual reality applications simulating real environments usually assume constant or piecewise-constant stiffness, stiffness can be an important continuous variable in data perceptualization [18]. For example, Yano et al. showed that a perceived stiffness distribution on a flat surface may not match its model and then proposed a simple compensation technique that prewarps the stiffness model [101]. Recently, Palmerius also proposed an effective haptization method for a stiffness map on a planar surface [79]. This algorithm adds tangential forces that contribute to accurate perception of stiffness changes during user's lateral scans, but its perceptual basis is still to be studied.

# Chapter 3

# Shape Modeling of Real Objects

In this chapter, we are concerned with how to model the shape of real objects, specially *soft, deformable* objects, easily and efficiently, for haptic rendering with a force-feedback haptic interface. The motivations behind this work are fourfold: (1) A force-feedback haptic interface is a high-resolution 3D position sensor; (2) Haptic modeling and rendering can be done with the same haptic interface, without requiring further, presumably expensive, equipment; (3) The rendering workspace of the interface is limited to the form factor which influence to the sensing range; and (4) A contact-based shape modeling approach with a force sensor can apply any object that is not appropriate for the optical scanner and be easily integrated into a force response modeling and a stiffness identification. The commercial MicroScribe share this philosophy, especially (1), but it can only deal with rigid objects. Our intent here is to develop a real object shape modeling software that can be easily used with general impedance force-feedback haptic interface for *haptic rendering* and a typical coordinate-measurement machine (such as the MicroScribe with an additional force sensor), even including soft objects which are inappropriate to capture an optical scanner, such as greasy biological tissues.

Our shape modeling system uses an ordinary force-feedback haptic interface with a probe including a force sensor (Sec. 3.1). The user grasps the probe and samples points on an object surface by tapping on the surface. Our sampling algorithm detects a contact using the force sensor and automatically compensates the position sampling error caused by the delay of contact detection (Sec. 3.2). The latter step is critical for soft, deformable objects. The 3D positions of the sampled points are processed to reconstruct a 3D mesh using the alpha-shape algorithm (Sec. 3.3). The 3D shape models created by our modeling system show acceptable quality compared with those made by a state-of-the-art optical 3D scanner. The appropriate uses and limitations of our system are also discussed (Sec. 3.4).

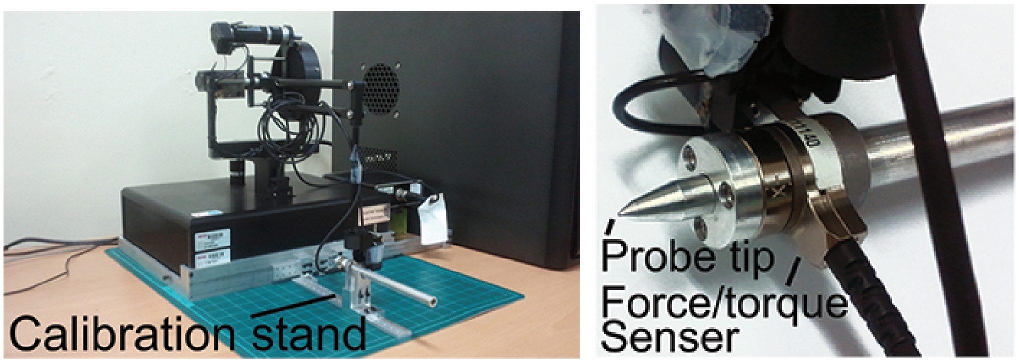
## 3.1 Hardware

Our current modeling system uses an impedance-type haptic interface (PHANToM Premium 1.5 High Force; Geomagic Technologies, Inc.) with a custom-made probe (Figure 3.1). For contact-based geometry modeling, it is ideal that the contact between a contactor and an object surface occurs at a point. Hence, a cone-shaped sharp tip is added to the end of the PHANToM's genuine encoder stylus (Fig. 3.1b). In our application, a contact-force modeling is quite small. Thus the sharp tip hardly ruptures or punctures on the object surface. In addition, a 6-axis force/torque sensor (Nano 17; ATI Industrial Automation, Inc.) is installed between the probe tip and the encoder stylus for contact detection. Note that a simple single-axis load cell is sufficient. We use the 6-DoF force sensor only because we have a working system from our previous research on haptic augmented reality [47, 48, 49]. The PHANToM is controlled by a PC on Windows XP at a sampling rate of 1 kHz. A calibration stand is used for more accurate and consistent sensing, although it is not absolutely necessary.

To use the modeling algorithms proposed in this thesis with a typical impedance-type haptic interface, the force-sensing probe tip needs to be fabricated, which is fairly straightforward and standard in haptics research.

## 3.2 Collection of Surface Points

Our shape modeling procedure begins with collecting the positions of 3D points on the surface of an object. This point set will be used for shape modeling described in the next



(a)PHANToM 1.5 High Force.

(b)Custom probe.

**Fig. 3.1** Haptic interface for shape modeling.

section.

For point sampling, the user holds the PHANToM stylus and taps on the object surface. Processing of each point consists of two steps. First, our modeling software detects a contact on the surface. Due to the digital nature, the contact detection occurs after the true contact. Second, we estimate the true contact time by backtracking the stored data. This step is the core of our algorithms designed to cope with the deformation in soft objects. Only the first step is sufficient for rigid objects.

### 3.2.1 Contact Detection

To detect a contact between the haptic interface probe and an object surface, we need to determine which variables to monitor from the interface. If a contact occurs, the object begins to exert force to the probe at an increase rate depending on the stiffness of the object. Applying a positive threshold to the magnitude of this external force is an easy and effective way for contact detection. This is the reason for using a force sensor in our system.

In most cases, raw force sensor readings are contaminated by other force components such as gravity and the inertial force of the contactor produced by the user's movements. They must be compensated for obtaining the external force exerted only by the object. Otherwise, measurements can suffer from excessive errors, and even false alarms (declaring a contact for non-contact free movements) can occur. Appropriate compensation procedures

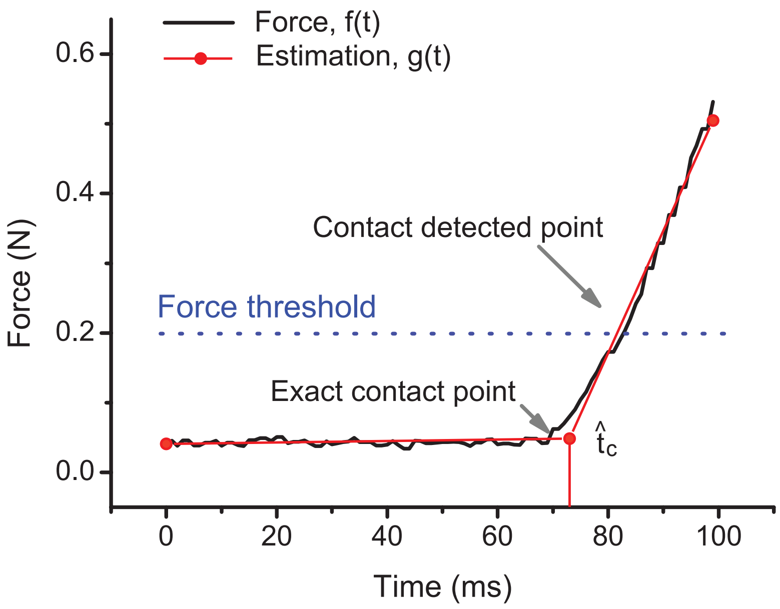


**Fig. 3.2** Measured force profile at a contact.

depend on the installation configuration of a force sensor and the kinematics of a haptic interface. In our system, we use the procedures described in [48] and obtain the contact force $\mathbf{f}(t)$. These procedures include estimation of the probe acceleration from the joint angles to remove the inertial force of the probe tip.

If $\|\mathbf{f}(t)\| > \varepsilon_f$ for a predetermined constant $\varepsilon_f > 0$, we declare a contact and proceed to the next step. An example force profile is shown in Fig. 3.2. We further improve the time error between detected and true contact times by the procedures discussed in following sections

### 3.2.2 True Contact Time Estimation

The threshold-based contact detection is straightforward, efficient, and appropriate for real-time algorithms. However, it can incur a large delay between detected and true contact times. This can make detected contact points deviate from true contact points, by a large degree for soft objects. Thus, we further estimate the true contact points from the stored data as follows.

The idea is simple. As physical contact is a discontinuous phenomenon, we should be able to find a knee point in the time graph of $\mathbf{f}(t)$. To find the knee point, we use a piecewise-linear regression algorithm presented in [12]. Define $f(t) = \|\mathbf{f}(t)\|$ over the interval $[t_{d-N}, t_d]$, where $t_d$ is the detected contact time by the thresholding and $N$ is a window size. We approximate $f(t)$ by two connected lines, $g(t)$, as shown in Fig. 3.2, such that

$$g(t) = \begin{cases} \dfrac{g_c(t - t_{d-N}) + g_{d-N}(t_c - t)}{t_c - t_{d-N}} & t_{d-N} \le t \le t_c \\ \dfrac{g_d(t - t_c) + g_c(t_d - t)}{t_d - t_c} & t_c \le t \le t_d \end{cases}, \tag{3.1}$$

where $t_c$ is the time index of the knee point of $g(t)$ and $g_{d-N}$, $g_c$ and $g_d$ are the values of $g(t)$ at $t_{d-N}$, $t_c$, and $t_d$, respectively.

For function fitting, we use the sum of square errors (SSE) between $f(t)$ and $g(t)$ as a cost function, such that

$$E = \sum_{i=t_{d-N}}^{t_d} (f(i) - g(i))^2. \tag{3.2}$$

This can be rewritten as

$$E(G, t_c) = \sum_{i=t_{d-N}}^{t_c} (f(i) - g(i))^2 dt + \sum_{i=t_c}^{t_d} (f(i) - g(i))^2. \tag{3.3}$$

where $G = [g_{d-N} \;\; g_c \;\; g_d]^{\mathrm{T}}$.

Therefore, the problem is formulated as: given $t_{d-N}$, $t_d$, and $f(t)$, find $\hat{t}_c$ and $\hat{G}$ such that $E(\hat{G}, \hat{t}_c)$ is minimized. We regard $\hat{t}_c$ as the true contact time and its probe tip position, $\mathbf{p}(\hat{t}_c)$, as the true contact point. Fig. 3.2 shows sample results of this estimation.

The following procedures are used to find $\hat{G}$ and $\hat{t}_c$ that minimize $E(\hat{G}, \hat{t}_c)$ [12]. First, the necessary conditions for the minimization are

$$\nabla_G E(G, t_c)|_{(\hat{G}, \hat{t}_c)} = 0 \quad \text{and} \quad \nabla_t E(G, t_c)|_{(\hat{G}, \hat{t}_c)} = 0. \tag{3.4}$$

To solve this, we formulate the closed form expression of $E(G, t_c)$. If $t_c$ is given, $E(G)$ can be expressed as

$$E(G) = \sum_{i=t_{d-N}}^{t_d} \left\{ f(i)^2 - 2B(i)^{\mathrm{T}} G + G^{\mathrm{T}} A(i) G \right\}, \tag{3.5}$$

where a $3 \times 3$ positive semi-definite matrix $A$ is

$$A = \begin{bmatrix} a_{11} a_{12} & 0 \\ a_{21} a_{21} & a_{23} \\ 0 a_{32} & a_{33} \end{bmatrix}, \tag{3.6}$$

$$\begin{aligned}
a_{11} &= \sum_{j=t_{d-N}}^{t_c} \frac{(t_c - j)^2}{(t_c - t_{d-N})^2}, \quad a_{12} = \sum_{j=t_{d-N}}^{t_c} \frac{(j - t_{d-N})(t_c - j)}{(t_c - t_{d-N})^2}, \\
a_{21} &= \sum_{j=t_{d-N}}^{t_c} \frac{(j - t_{d-N})(t_c - j)}{(t_c - t_{d-N})^2}, \\
a_{22} &= \sum_{j=t_{d-N}}^{t_c} \frac{(j - t_{d-N})^2}{(t_c - t_{d-N})^2} + \sum_{j=t_c}^{t_d} \frac{(t_d - j)^2}{(t_d - t_c)^2}, \\
a_{23} &= \sum_{j=t_c}^{t_d} \frac{(t_d - j)(j - t_c)}{(t_d - t_c)^2}, \quad a_{32} = \sum_{j=t_c}^{t_d} \frac{(t_d - j)(j - t_c)}{(t_d - t_c)^2}, \\
a_{33} &= \sum_{j=t_{d-N}}^{t_c} \frac{(j - t_c)^2}{(t_d - t_c)^2},
\end{aligned}$$

and a $3 \times 1$ column vector $B$ is

$$B = \begin{bmatrix} c \displaystyle\sum_{j=t_{d-N}}^{t_c} f(j) \frac{t_c - j}{t_c - t_{d-N}} \\ \displaystyle\sum_{j=t_{d-N}}^{t_c} f(j) \frac{j - t_{d-N}}{t_c - t_{d-N}} + \sum_{j=t_c}^{t_N} f(j) \frac{t_N - j}{t_N - t_c} \\ \displaystyle\sum_{j=t_c}^{t_N} f(j) \frac{j - t_c}{t_N - t_c} \end{bmatrix}. \tag{3.7}$$

The necessary and sufficient condition for $E(G)$ to achieve a minimum at $\hat{G}$ is

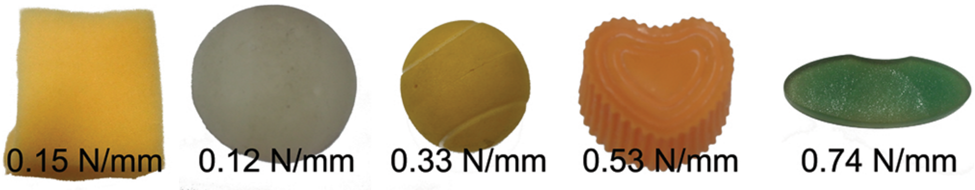
$$\left. \frac{\partial E(G)}{\partial G} \right|_{G=\hat{G}} = 0. \tag{3.8}$$

Therefore, $\hat{G}$ can be computed by

$$\hat{G} = A^{-1} B. \tag{3.9}$$

Then, the necessary condition (3.4) can be rewritten as

$$\nabla_t E(\hat{G}(t_c), t_c) \big|_{\hat{t}_c} = 0. \tag{3.10}$$



(a)Sponge block (b)Silicone hemisphere (c)Foam ball (d)Ridged silicone (e)Silicone rest

**Fig. 3.3** Sample objects used in the evaluation. Their stiffness shown together was measured when applied force was 5 N.

Therefore, the SSE to be minimized becomes

$$E_{\hat{G}}(t_c) = E(G,t_c)|_{G=\hat{G}(t_c)}\,. \tag{3.11}$$

This means that we can $\hat{t_c} \in [t_{d-N}, t_d]$ such that $E_{\hat{G}}(\hat{t_c})$ is minimized. As $N$ is relatively small, the optimal $\hat{t_c}$ can be found even with a simple brute-force search. Our current implementation use the downhill simplex method [71]. This method is a common nonlinear optimization technique and iterates evaluating a cost function at the vertices of a simplex and then replacing the worst point of the simplex by a new point until a desired bound is obtained.

### 3.2.3 Performance Evaluation

We evaluated the performance of the contact point estimation algorithm with five real soft objects shown in Fig. 3.3. In the test, each object was fixed to a vertical wall placed in front of the PHANToM. The probe moved on a rail so that it always collided with an object at the same point. The experimenter carefully selected a true contact point by monitoring the force sensor output. Then, the experimenter repeatedly tapped on the object on the rail to collect 100 contact points. The parameters used were: $\varepsilon_f = 0.2$ N and $N = 100$. In this way, each contact point could be compared with the true contact point.

Experimental results are summarized in Fig. 3.4 where two kinds of errors are shown for each object. The first error is the distance between a contact point found by the simple thresholding and a true contact point. The other error is for a contact point resulted from

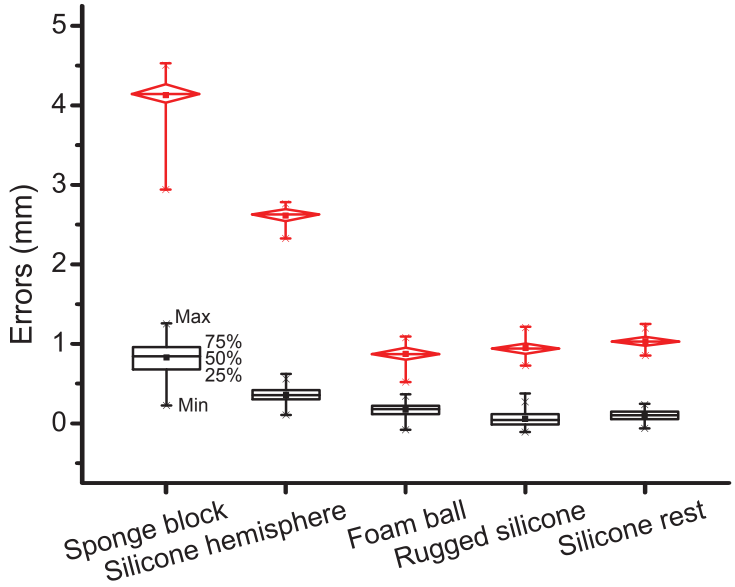
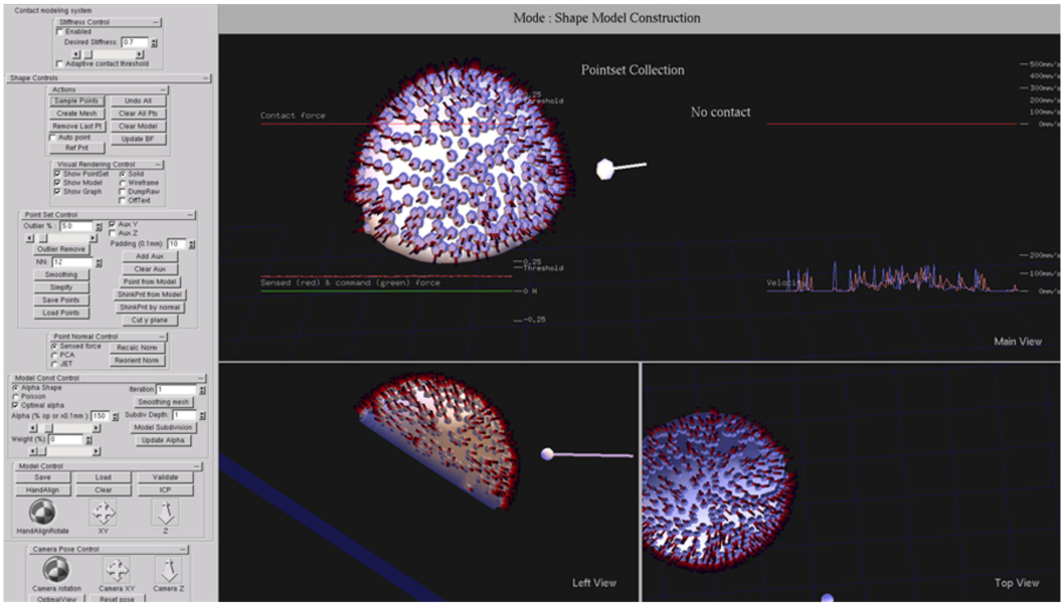


**Fig. 3.4** Distance errors between estimated contact points and true contact points on an object surface. Results of the simple thresholding are represented by diamond box plots, and those of the true contact point estimation algorithm are by box plots.

our true contact point estimation algorithm. All the results demonstrate that the true contact point estimation algorithm greatly reduced the position errors of the simple thresholding. The estimation errors were less than 0.5 mm except for the sponge block, and this accuracy is comparable to that of the optical tracker. This demonstrates that our algorithm allows to restore the original contact points to an accuracy comparable to the laser scanner for moderately deformable objects. For the sponge block that we selected as the worst case, the estimation error was about 1.5 mm. This soft object even shows some plasticity that delays the restoration of a deformed part, and this seems to be responsible for the increased error.

## 3.3 Modeling of Surface Points

Our system reconstructs a surface model from the surface points collected by the procedures described in the previous section. For this purpose, the geometry construction algorithms for 3D optical scanners can be adapted since both use a point set to generate surface meshes.


Contact modeling system

Stiffness Control: □ Enabled; Desired Stiffness: 0.7; □ Adaptive contact threshold

Actions: □ Auto point

Visual Rendering Control: ☑ Show PointSet ☑ Show Model ☑ Show Graph; ● Solid ○ Wireframe; □ DumpRaw □ OffText

Point Set Control: Outlier % : 5.0; ☑ Aux Y □ Aux Z; Padding (0.1mm): 10; NN: 12

Point Normal Control: ● Sensed force ○ PCA ○ JET

Model Const Control: ● Alpha Shape ○ Poisson; ☑ Optimal alpha; Alpha (% op or x0.1mm ): 150; Weight (%): 0; Iteration 1; Subdiv Depth: 1

Mode : Shape Model Construction

Pointset Collection

No contact

**Fig. 3.5** Modeling program. All of point set processing and surface reconstruction are supported by GUI.

In this section, we describe the computational procedures for mesh reconstruction used in our system and report modeling results in comparison with a 3D optical scanner.

### 3.3.1 Reconstruction Procedures

While hundreds of thousands of surface points can be sampled using a 3D optical scanner, our contact-based sampling procedures allow a smaller number of point samples (usually thousands of points) due to the need of manual tapping. Therefore, our reconstruction procedures are designed taking into account this characteristic.

Various point reconstruction algorithms have been studies in computer graphics [6]. The majority of them make use of an implicit surface constructed from a point set as an intermediary step [51]. These algorithms are robust to sampling noise, which is enabled by the implicit surface approximation. However, if the number of point samples is insufficient, the implicit surface approximation can be inaccurate, and this can cause a large modeling error in the final mesh model. Thus, this approach is not appropriate for our system that relies on relatively sparse sampling.

In our current modeling system, we use the alpha-shape algorithm [29, 4], which is based on Delaunay triangulation for surface reconstruction [6]. In particular, this algorithm uses an input point set without any approximation and shows good reconstruction performance if the input point set is noise-free and uniformly sampled. In our system, the user has a full control over the positions of sample points and can use their best judgements to maintain the uniform spatial density of samples. The system also provides a user interface (Fig. 3.5) to aid sampling, as well as instant removal of erroneous samples.

Before the reconstruction, we preprocess the input point set for noise reduction. First, for each point, its $k$-nearest neighbors are found, and the average distance between the point and the nearest neighbors are computed. The points with abnormally large distances are considered as outliers and discarded. Second, for each point, we find a best-approximating local surface around the point by fitting a smooth second-order parametric surface patch to its $k$-nearest neighbors [13]. The input point is then projected onto the parametric local surface patch for further smoothing.

The alpha shape algorithm is performed on this preprocessed point set. The algorithm first performs Delaunay triangulation to find the convex hull of an input point set and the simplex (e.g., triangle or tetrahedron) of each input point. The algorithm includes the simplex in surface reconstruction if the radius of the circumsphere of the simplex is smaller than $\alpha$ and the circumsphere does not include other points. The resulting surface mesh only includes points the simplex of which is “smaller” than $\alpha$. Thus, $\alpha$ corresponds to a resolution of the model and determines the details of modeling; $\alpha = 0$ results in the individual points and $\alpha = \infty$ the convex hull. The user can set $\alpha$ on their own, or use the value that generates one connected component, which can be found automatically [4].

Output meshes of the weighted alpha shape algorithm can still can be noisy and coarse. Thus, we further perform two standard algorithms, Taubin smoothing [93] and $\sqrt{3}$ subdivision [54], to generate smooth and dense meshes. Taubin smoothing can attenuate noise while preserving the surface, and $\sqrt{3}$ subdivision is best suited for triangular meshes. Our implementation of the above geometry operations makes use of the CGAL computational geometry library [31].

### 3.3.2 Performance Evaluation

To evaluate the performance of our modeling system, we modeled the same real objects using a commercial 3D optical scanner and our contact-based system. The optical scanner (Rexcan3, Solutionix corp.) uses twin cameras and a white-light phase-shifting triangulation technique, with a scan resolution 0.04–0.48mm. Those captured using this optical scanner can be considered as models of the highest accuracy. Our true contact point estimation and simple thresholding were used for our contact-based system. Three real objects, the foam ball (Fig. 3.3c), the ridged silicone (Fig. 3.3d), and the silicone hemisphere (Fig. 3.3b) were used for comparison.

Measurement using the 3D scanner took around 15 minutes per a object. Some of the objects had to be coated with a 3D scan spray or a powder spray because of reflection and/or transparency. The collected points were simplified to tens of thousands of points, and then smoothed. The alpha shape algorithm was used to reconstruct a surface model for the foam ball and the silicone hemisphere. We also used the Poisson surface reconstruction for the ridged silicone to preserve its jagged detail. On the other hand, point sampling using our tapping method required about 5–15 minutes depending on the object. Then, mesh models were reconstructed following the procedures described in the previous section. $\alpha$ was determined using [5] so that the models form a single connected component. Modeling results are presented in the first, second, and third rows of Fig. 3.6.

We also analyzed the modeling errors quantitatively. For this, we first aligned the meshes obtained by the 3D scanner and our system using the iterative closest point algorithm [7]. Then, they were cropped to have the same region for comparison. Lastly, we computed

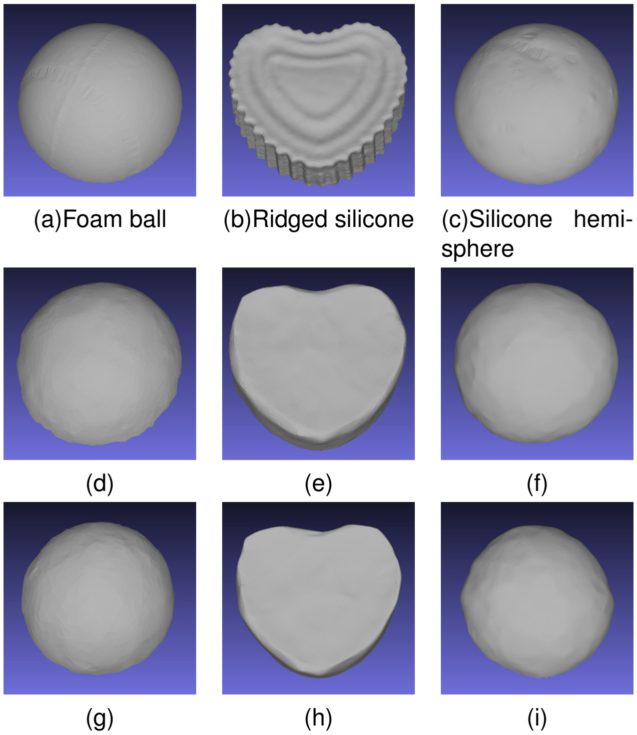


**Fig. 3.6** Examples of reconstructed models from soft real objects. (a), (b), and (c) were scanned by the optical 3D scanner. The models in second and third rows were captured and built by our modeling system. The points of (d), (e), and (f) were estimated by our point estimation algorithm. For (g), (h), and (i), simple thresholding algorithm was used.

the Hausdorff distance[1] between the common part of the two meshes using the Metro tool kit [19].

Table 3.1 summarizes the results of the quantitative error analysis.

When we used the true contact point estimation, the Hausdorff distances of the foam ball and the silicone hemisphere were around 2–3 mm. Since the Hausdorff distance is a max-

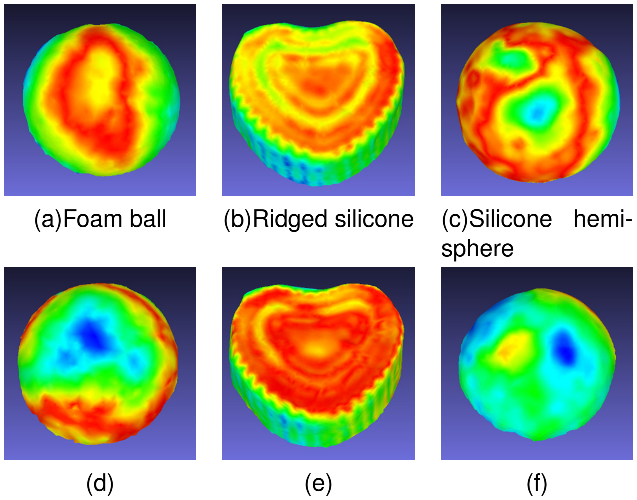
[1] The Hausdorff distance between surfaces $S_1$ and $S_2$ is

$$E_H(S_1, S_2) = \max\{E(S_1, S_2), E(S_2, S_1)\}, \tag{3.12}$$

where

$$E(S_i, S_j) = \max_{p \in S_i} \min_{p' \in S_j} d(p, p'). \tag{3.13}$$

This distance is used as a standard metric for the difference between two geometric representations.



(a)Foam ball (b)Ridged silicone (c)Silicone hemisphere

(d) (e) (f)

**Fig. 3.7** Error distributions of reconstructed models. The presentation used a RGB color map with red and blue indicating minimum and maximum, respectively. (a), (b), and (c) captured by our modeling system with the true contact point estimation algorithm. (d), (e), and (f) built by our modeling system with simple thresholding. with simple thresholding.

min criterion, average distances between models were shorter than 1 mm. For the ridged silicone, the Hausdorff distance was increased to 4.9 mm, because the ridged part was not captured by our modeling system.

With simple thresholding, the Hausdorff distance of the silicone hemisphere was 6.7 mm, and this result is remarkably increased from the result with the true contact point estimation From this result, our true contact point algorithm was works well and effective for the objects of low stiffness. For the foam ball and the ridged silicone, the Hausdorff distances were similar to the results with the true contact point estimation. In this case, two models have relatively high stiffness. Thus the errors originated from the contact point were decreased. During the reconstruction stage, filtering canceled these small position estimation errors.

The error distributions in Fig. 3.6 show the detailed errors of on the surfaces. It shows high deviations at the ridged region.

**Table 3.1** Quantitative comparisons of modeling performance. O: Optical scanner. P: Our probe-based method. S: A probe-based method with simple thresholding.

| | Foam ball | Ridged silicone | Silicone hemisphere |
|---|---|---|---|
| Num. of sampled points (O) | 33,489 | 30,900 | 30,000 |
| Num. of vertices (O) | 3,493 | 12,211 | 12,742 |
| Num. of faces (O) | 6,801 | 26,076 | 25,115 |
| Num. of sampled points (P) | 1,004 | 943 | 296 |
| Num. of vertices (P) | 1,195 | 3,822 | 3,196 |
| Num. of faces (P) | 2,286 | 7,488 | 6,239 |
| Hausdorff distance (O and P) (mm) | 2.16 | 4.92 | 3.30 |
| Num. of sampled points (S) | 1,004 | 943 | 296 |
| Num. of vertices (S) | 961 | 3,267 | 2,675 |
| Num. of faces (S) | 1,820 | 6,401 | 5,189 |
| Hausdorff distance (O and S) (mm) | 2.82 | 4.67 | 6.69 |

## 3.4 General Discussion

Our contact point estimation algorithm based on the response force showed the error range comparable to the 3D optical scanner. For common elastic objects, the average estimation errors and the min-max errors were smaller than 0.5 mm. These contact point set can be used for point cloud haptic rendering algorithms [85, 62].

In terms of the final shape modeling accuracy, our modeling results showed 2–3 mm of Hausdorff distance error in comparison with the state-of-the-art 3D optical scanner. Especially, our modeling system with the contact point estimation algorithm effectively reduces the surface reconstruction error of low-stiffness model. The Hausdorff error was increased to about 5 mm for the object containing the fine details that were difficult to be captured with manual tapping. If we upgrade our system to be able to sample surface points by automatic tapping and collect a larger number of points, we may expect further improvements in modeling accuracy.

Our modeling results can also be assessed in terms of perception, e.g., using the human haptic discriminability of size. For instance, Durach et al. showed that the JND of length perception was roughly 1 mm for the reference lengths of 10 to 20 mm in a real environment [28]. O'Malley and Goldfarb used rounded ridges of the length of 10 to 20 mm for size discrimination [53]. The participants correctly identified a size difference of 2.5 mm with about 60% chance in a virtual environment. The Hausdorff distance errors shown for regular elastic objects were comparable to these size discrimination JND, indicating that in practice it would be difficult for the user to perceive the modeling errors by hand.

Our modeling system can produce the shape models that have sufficient accuracy and resolution for haptic rendering. Hence, the system is the most appropriate for the cases where no visual counterparts are necessary, e.g., for haptic augmented reality where real objects are seen as they are but their haptic properties are augmented [47, 48]. When high quality visual rendering is also desired, our method can provide a rapid baseline model to authoring software, from which the user can refine the details, accelerating the modeling process.

One of a primal use of our system is the basis part of the force response modeling for haptic rendering. The contact estimator can also contribute to improving the accuracy of modeling the contact dynamics of a real object by providing more exact deformation of the object.

Chapter 4

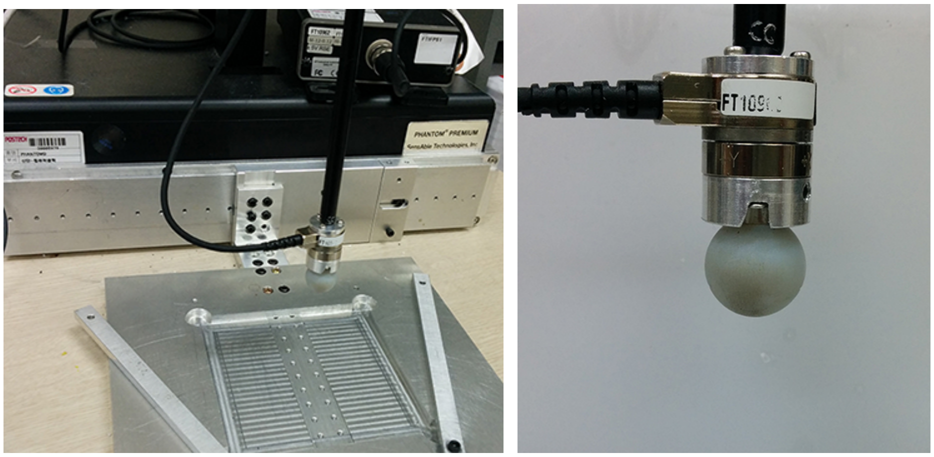
# Force Response Modeling of Real Objects

In this chapter, the modeling system of the force response of inhomogeneous real objects. A data-driven approach is adopted to model the complex force response behaviors. With the shape modeling framework introduced in the previous chapter, we can model and render the real objects.

Our modeling system can model the real objects that satisfy the following assumption. We only consider the real objects of moderate stiffness with no plasticity. The plasticity in the object makes highly complex force response to be modeled. High stiffness of the object also cannot be handled due to limited sensing ability of our system configuration.

## 4.1 Hardware

Our current modeling system uses an impedance-type haptic interface (PHANToM Premium 1.5 High Force; Geomagic Technologies, Inc.) with a spherical probe (diameter: 1.6 cm, Fig. 4.1). In addition, a 6-axis force/torque sensor (Nano 17; ATI Industrial Automation, Inc.) is installed between the probe tip and the end-effector of the haptic interface. Then, we can control the probe center position automatically. The PHANToM is controlled by a PC on Windows XP at a sampling rate of 1 kHz. A jig is used to fix the sample and it is



(a)PHANToM 1.5 High Force with Jig.

(b)Custom probe with force sensor and spherical probe(diameter: 1.6 cm)

**Fig. 4.1** Haptic interface for modeling the force responses.

also used as a reference frame of the calibration. The spherical probe can have a consistent contact shape for arbitrary position. The maximum continuous force output of our haptic interface is 6.2 N. Thus, our modeling range of the force is limited to 5 N.

## 4.2 Interpolation Model

We introduce a new interpolation model for describing the contact between the probe tip and the soft object. When the probe tip and the soft object contact, the external force is applied to the target object through the probe tip, and deforms the shape of the target. The reaction force is also exerted on the probe tip surface through the contact area. The estimated reaction force is calculated as following interpolation model,

$$\tilde{\mathbf{f}} = \mathbf{g}(\mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}), \tag{4.1}$$

where $\mathbf{p}$ is the probe center position, $\dot{\mathbf{p}}$ is the velocity of the probe center position and, $\mathbf{q}$ is proxy, the return point. These points are represented at Fig. 4.2.

The probe center position is located on the surface of the target object, and it can represent the position of the deformed surface. The return point is the probe center point when

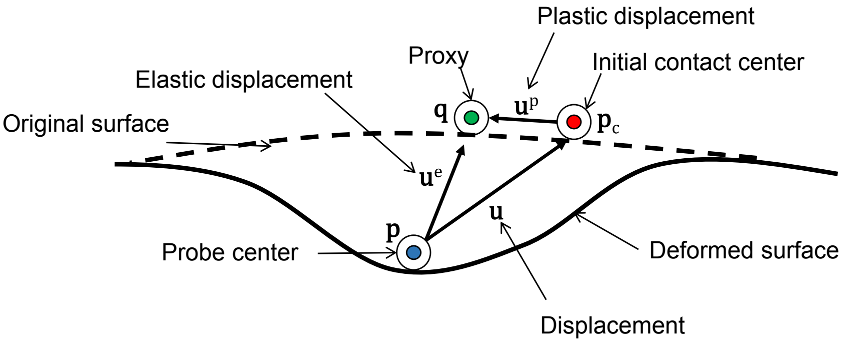


**Fig. 4.2** Important points and vectors for interpolation model .

the shape of elastic target object returns its original shape. If the external force returns zero, the deformed surface is restored to its original surface, and the probe contacting with the surface also moves to the outside of the original surface. With the probe center, this proxy can represent the elastic deformation of the object, because the probe center elastically returns to the proxy.

### 4.2.1 Description of Proxy

We introduce the proxy using the analogy of the elasto-plasticity. In our model, $\mathbf{p}$ and $\dot{\mathbf{p}}$ are used to describe current states of the probe. However, the description of the proxy is not trivial and it required to additional explanation.

We considered the interaction between the probe and object as an analogy of loading and unloading process in the elasto-plasticity (See Fig. 4.3) [69, 23, 36]. Loading is increasing of the stress applying on the object and unloading is the decreasing of the stress. In the typical elasto-plastic case, the strain, the constitutive counter part of stress, expressing the deformation, is increased in the loading condition, but the only elastic part of strain is decreased in the unloading condition. The plastic part of strain occurs when the loading is processed and the stress the stress reaches a certain elastic limit, a yield point. If the unloading is completed, the elastic part of strain becomes zero but the plastic part of strain is remained. This means the elasto-plastic material returns its plastically deformed shape.

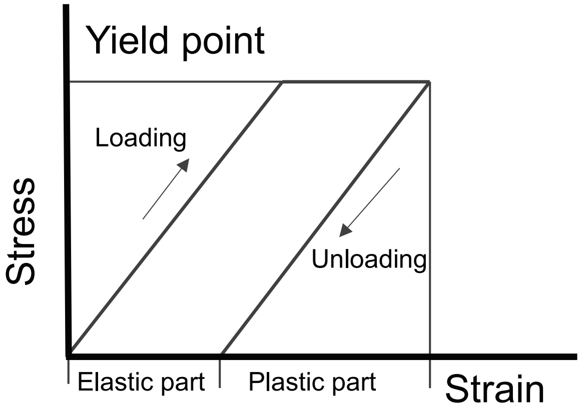


**Fig. 4.3** Loading and unloading processes in the elasto-plasticity

Similar analogy can be applied to the interaction between the probe and the object. The probe center position **p** is located on the surface of the object in the contact. Then, a compressive external force can be applied from the probe to the object and it can deform the object shape. We denote the compressive external force to deform the object shape as the contact load force $\mathbf{f}_l$. Then, we denote the probe center position at the initial contact as $\mathbf{p}_c$. This point is the probe center position when the contact is just initiated, and the $\mathbf{f}_l$ is zero. Then, we can define the displacement **u** of the probe center as

$$\mathbf{u} = \mathbf{p} - \mathbf{p}_c. \tag{4.2}$$

This displacement vector and its relating vectors are illustrated in Fig.4.2. Now, we have a pair of the **u** and $\mathbf{f}_l$.

As the elasto-plasticity analogy, we will call the increasing of $\mathbf{f}_l$ as contact loading and the decreasing of $\mathbf{f}_l$ as contact unloading. Then, we can make the constitutive relation between the **u** and $\mathbf{f}_l$. In the contact loading process, the $\mathbf{f}_l$ is increased and has non-zero value. Then, the soft object is deformed, and the probe center position **p** located on the deformed surface, has been changed. As the contact loading process proceeds, **p** moves away from the $\mathbf{p}_c$. Thus, the magnitude of **u** is also increased. In the contact unloading process, the $\mathbf{f}_l$ is decreased. Then, the deformed object surface is returned its original shape, and the probe tip on the deformed surface is also moved to the outside of the original surface. Thus, the $\mathbf{f}_l$ is returned to zero, the probe tip is located somewhere on the original surface. However, this returned probe center can be different from $\mathbf{p}_c$, and $\mathbf{u}$ can have non-zero value even if $\mathbf{f}_l$ is zero. We can consider this remained part of $\mathbf{u}$ as a plastic displacement $\mathbf{u}^p$.

The plastic displacement is originated from sliding interaction between the probe and the deformed surface when the contact is initialized. If there are no relative displacement between the initial contact points, $\mathbf{p}$ returns $\mathbf{p}_c$. However, if the relative displacement in the contact area is occurred, $\mathbf{p}$ should not be return $\mathbf{p}_c$. The condition occurring the relative displacement is related to the frictional behaviors. Then, the other portion of $\mathbf{u}$ is considered as an elastic displacement $\mathbf{u}^e$, because this portion is returned and zero when $\mathbf{f}_l$ returns zero. This elastic displacement represents the displacement from the current probe point to the outside of the original surface of the object.

Now, we can apply the separation of elastic and plastic quantity of elasto-plasticity to the displacement. The displacement can be separated into two vectors as,

$$\mathbf{u} = \mathbf{u}^e + \mathbf{u}^p, \tag{4.3}$$

for any $\mathbf{f}_l$. This means that we can express the current plastic and elastic displacements in any state in the contact loading and unloading. Thus, we can handle the sliding interaction as the plastic displacement.

Then, we can express the proxy position $\mathbf{q}$ as

$$\mathbf{q} = \mathbf{p}_c + \mathbf{u}^p, \tag{4.4}$$

for any $\mathbf{f}_l$. This proxy is always located on the outside of the original shape of the object. We can consider the proxy point as the plastic change from the initial contact.

### 4.2.2 Estimating of Input Parameters

In order to train our interpolation model, we should measure the reaction force $\mathbf{f}$ and relating input variables, $\mathbf{p}$ and $\dot{\mathbf{p}}$ and estimate $\mathbf{q}$. These variables should be measured concurrently. Measuring $\mathbf{f}$ and $\mathbf{p}$ is trivial when we use the force sensor and encoder. $\dot{\mathbf{p}}$ can be estimated using velocity filter from **p**. However, **q** is only measured explicitly when the external loading force is zero, and thus, it cannot be measured concurrently with **f**, **p**, and $\dot{\mathbf{p}}$. Thus, we should estimate **q** by using additional preprocessed information. Now, we will explain procedures to estimate the proxy.

As equation 4.3, **q** is determined by $\mathbf{u}^p$ from $\mathbf{p}_c$. $\mathbf{p}_c$ can be estimated using the contact detection between the probe and the object. We use geometrical contact detection algorithm. [34] using the shape model for contact detection by measuring the shape model of the original surface. This shape model is measured by contact measurement algorithm introduce at 3 with spherical probe tip. Then, we have unknown $\mathbf{u}^p$ and known **p**, and $\mathbf{p}_c$, **f**. From these known parameters, we also have **u**. In order to estimate $\mathbf{u}^p$, we should decompose **u** into $\mathbf{u}^p$ and $\mathbf{u}^e$. A sliding yield surface originated from the the elasto-plastic analogy of the frictional phenomenon is used in the decomposition phase.

The detailed information of the sliding yield surface including the definition and capturing method are represented at Sec. 4.3. The procedures to decompose the displacement and estimate the proxy are also represented.

Then, if the input parameters can be estimated, the interpolation model should be trained. The radial based function is used to describe the relation between the input parameters and the measured force. The detailed information about training of interpolation function and calculating rendering force is represented at Sec. 4.4.

### 4.2.3 Roles of Modeling Variables

Our interpolation model is designed to describe the elastic information using two variables, **p** and **q**. **p** is located on the deformed surface and **q** is located on the original surface. Then, the elastic displacement $\mathbf{u}^e$ can be expressed as

$$\mathbf{u}^e = \mathbf{q} - \mathbf{p}. \tag{4.5}$$

Moreover, including **q** allows us to interpolate the information during the sliding.

Previous approaches used a single variable, **u** to describe the elastic information. If the plastic displacement is zero, $\mathbf{u}_e$ is equal to **u**. However, if there is non-zero plastic displacement, we must find $\mathbf{u}^e$ explicitly. Moreover, using single variable $\mathbf{u}$ is only valid for given $\mathbf{p}_c$, and it required to train the data-driven model for each given $\mathbf{p}_c$. In order to describe the arbitrarily $\mathbf{p}_c$, it required interpolate the estimation of the data-driven model.

Our interpolation model includes also the velocity of the probe $\dot{\mathbf{p}}$ to describe viscous information of the object. This information is very limited to describe the viscosity, but to endure simplicity of the data-driven approach, we only use single variable $\dot{\mathbf{p}}$.

## 4.3 Sliding Yield Surface

As illustrated at section 4.2.1, we consider the contact as the contact loading and unloading process. The frictional phenomenon can be interpreted as a plastic deformation. A sliding yield surface, a surface in the space of the loading force $\mathbf{f}_l$, is introduced to describe the frictional limit. If the loading force is inside the sliding yield surface, $\mathbf{u}$ is considered elastic displacement $\mathbf{u}^e$. When the loading force is reached the sliding yield surface, $\mathbf{u}$ has plastic displacement $\mathbf{u}^p$. Then, even if the force is unloaded, and is inside the sliding yield surface, the plastic displacement $\mathbf{u}^p$ pertains in $\mathbf{u}$.

This sliding yield surface is derived from a yield surface. Yield surface is a border in the stress space which loading or reloading makes an elastic strain. Thus, if the applied stress to the object is inside of the yield surface, deformation can be considered as elastic deformation. This yield surface is also used to describe the constitutive relation of stress and strain.

For our interaction case, we define the sliding yield surface of the loading force $\mathbf{f}_l$ for given initial contact center $\mathbf{p}_c$ as

$$f_s(\mathbf{f}_l) = 0. \tag{4.6}$$

So, we can call the initial contact center $\mathbf{p}_c$ for sliding yield surface as a reference point $\mathbf{c}$.

We also can consider the loading force $\mathbf{f}_l$ is same as the measured force $\mathbf{f}$ by the law of action-reaction. This sliding yield surface has close relationship with the frictional phenomenon. Thus, we split $\mathbf{f}_l$ into normal and tangential components, $\mathbf{f}_N$ and $\mathbf{f}_T$, respectively.

This force also can be represented as,

$$\mathbf{f}_l = \mathbf{f}_N + \mathbf{f}_T, \tag{4.7}$$

$$= f_N \mathbf{n}_c + \mathbf{f}_T, \tag{4.8}$$

$$= f_N \mathbf{n}_c + f_T \mathbf{t}_c. \tag{4.9}$$

where $\mathbf{n}_c$ is the surface normal vector at $\mathbf{c}$, $\mathbf{t}_c$ is tangential unit vector, $f_N$ is the magnitude of the normal loading force, $f_T$ is the magnitude of tangential loading force. Using these normal and tangential components, we assume the sliding yield surface as

$$f_s(\mathbf{f}_l) = f_s(f_N, \mathbf{f}_T) = f_N - f_\mu(f_N, \mathbf{f}_T/||\mathbf{f}_T||)f_N \leq 0. \tag{4.10}$$

This surface shows the coefficient between the normal loading force and the tangential loading force. The coefficient function $f_\mu()$ can have diversity for the directions and the magnitude of the normal force. Then, we can determine $f_\mu()$ using the measured data. In section 4.3.2, the procedures to estimate $f_\mu$ will be represented.

For $\mathbf{f}_l$ satisfying $f_s$, we can also find $\mathbf{u}$ in the elastic limit from the initial contact center. Then, we also define the elastic limit of the displacement for given $\mathbf{p}_c$ as a constitutive counterpart of $f_s$,

$$g_s(\mathbf{u}) = 0. \tag{4.11}$$

This elastic limit can be used to determine the elastic displacement $\mathbf{u}^e$ from $\mathbf{u}$. This displacement space version of the sliding yield surface can be also called as sliding yield surface. However, in order to prevent confusion, we will call it as a elastic displacement limit surface.

### 4.3.1 Estimating Proxy Using Sliding Yield Surface

Estimating proxy position is based on the sliding yield surface. If the contact loading force is reached the sliding yield surface, the proxy should be updated. Then, the next issue is determining the position of the updated proxy. In order to determine it, we use that the current elastic displacement should be inside of the elastic displacement limit surface, and the elastic displacement limit surface can be derived from the measured information. Then, for each iteration, we can check that the contact loading force is inside of the sliding yield surface, and the current estimated elastic displacement is inside of the elastic limit of the displacement. If these conditions are violated, we update the proxy position, which should not violate the condition.

The steps of estimating proxy are iterative as follows. We denote current time step as $n$. At the initial contact frame, we have $\mathbf{p}_c$ and assign $n = 0$, $\mathbf{u}(0) = \mathbf{0}$, and $\mathbf{q}(0) = \mathbf{p}_c$. For each time step, the deference of $\mathbf{u}$ is denoted as $\Delta\mathbf{u}$.

$$\Delta\mathbf{u}(n) = \mathbf{u}(n) - \mathbf{u}(n-1). \tag{4.12}$$

Then, we also can decompose $\Delta\mathbf{u}(n)$ into elastic part, $\Delta\mathbf{u}^e(n)$ and the plastic part, $\Delta\mathbf{u}^p(n)$.

$$\Delta\mathbf{u}(n) = \Delta\mathbf{u}^e(n) + \Delta\mathbf{u}^p(n). \tag{4.13}$$

$\mathbf{u}(n)$, $\mathbf{u}(n)^e$, and $\mathbf{u}(n)^p$ can be represented in cumulative form as follows,

$$\mathbf{u}(n) = \sum_{i=0}^{n} \Delta\mathbf{u}(i), \tag{4.14}$$

$$\mathbf{u}^e(n) = \sum_{i=0}^{n} \Delta\mathbf{u}^e(i), \tag{4.15}$$

$$\mathbf{u}^p(n) = \sum_{i=0}^{n} \Delta\mathbf{u}^p(i). \tag{4.16}$$

If the $\mathbf{f}_l(n)$ is $f_s(\mathbf{f}_l) < 0$, $\Delta\mathbf{u}(n)$ is elastic. However, $\mathbf{f}_l(n)$ is reached $f_s(\mathbf{f}_l) = 0$, $\Delta\mathbf{u}(n)$ has the plastic part.

For the simplicity, we use the $\mathbf{u}(n)$ and $g_s(\mathbf{u}(n))$ instead of $\mathbf{f}_l(n)$ and $f_s$ to determine $\mathbf{u}^p(n)$ and $\mathbf{u}^e(n)$. In figure 4.4, basic schemes of decomposition are represented. Assume that $\mathbf{u}^p(n-1) = \mathbf{0}$ and $\mathbf{u}^e(n-1) = \mathbf{u}(n-1)$. Then, at $n$ frame, we can get the trial displacement $\mathbf{u}^*$ as

$$\mathbf{u}^*(n) = \mathbf{u}^e(n-1) + \Delta\mathbf{u}(n) \tag{4.17}$$

$$= \mathbf{u}^e(n-1) + \Delta\mathbf{u}^e(n) + \Delta\mathbf{u}^p(n), \tag{4.18}$$

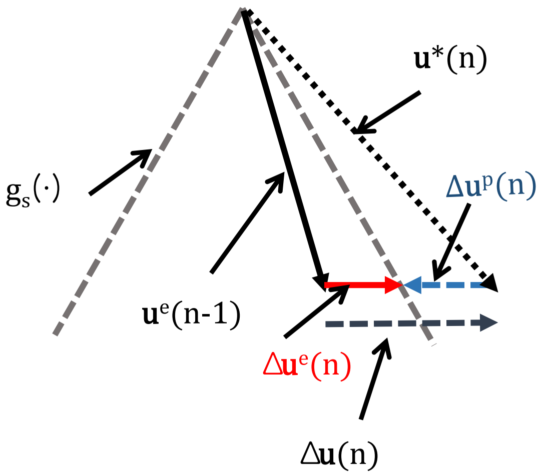


**Fig. 4.4** Decomposition of **u** using $g_s(\cdot)$

If $\mathbf{u}^*(n)$ is located the outside of $g_s$ for given $\mathbf{p}_c$, $\mathbf{u}^e(n)$ must be returned inside of $g_s$, as

$$g_s(\mathbf{u}^e(n)) = 0, \tag{4.19}$$

$$g_s(\mathbf{u}^e(n-1) + \Delta\mathbf{u}^e(n)) = 0. \tag{4.20}$$

Then, we can calculate $\Delta\mathbf{u}^e(n)$ from $\mathbf{u}^e(n-1)$ and the condition $g_s(\mathbf{u}^e(n)) = 0$ for given $\mathbf{p}_c$. Using these variables, we also estimate $\Delta\mathbf{u}^p$ and $\mathbf{u}^p(n)$.

Finally, we can determine $\mathbf{q}(n)$ as

$$\begin{aligned} \mathbf{q}(n) &= \mathbf{u}^p(n) + \mathbf{p}_c & (4.21)\\ &= \mathbf{u}^p(n-1) + \Delta\mathbf{u}^p(n) + \mathbf{p}_c & (4.22)\\ &= \mathbf{q}(n-1) + \Delta\mathbf{u}^p(n). & (4.23) \end{aligned}$$

Then, the proxy is updated, $f_s(\cdot)$ and $g_s(\cdot)$ must be also updated (See figure 4.5). $f_s(\cdot)$ and $g_s(\cdot)$ are valid for given $\mathbf{p}_c$. If the loading force is $f_s(\cdot)$, the probe is returned to $\mathbf{p}_c$. However, if the loading force is outside the $f_s(\cdot)$, and the proxy is updated, the probe is not returned to $\mathbf{p}_c$. Then, $f_s(\cdot)$ and $g_s(\cdot)$ should be updated for new return point, the proxy.

Thus, we should model $f_s(\cdot)$ and $g_s(\cdot)$ of the interest region of the target object, because $f_s(\cdot)$ and $g_s(\cdot)$ are depended on the initial contact center. In order to generate $f_s(\cdot)$ and $g_s(\cdot)$

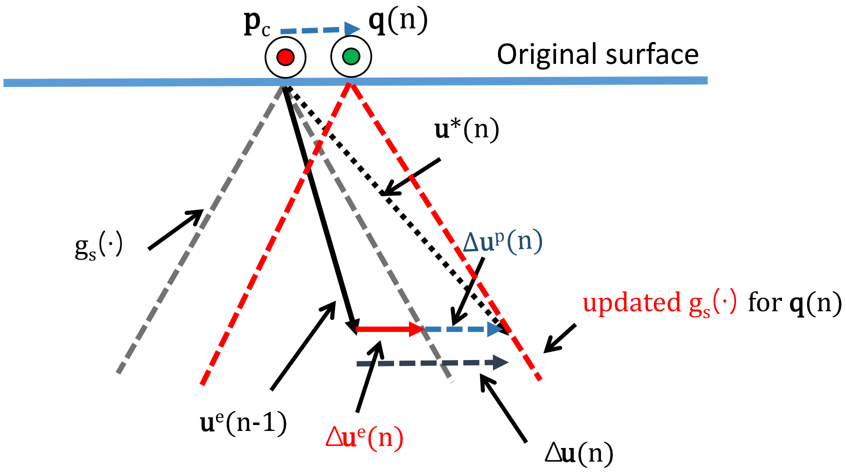


**Fig. 4.5** Update of $g_s(\cdot)$

for arbitrary initial contact center, the interpolation scheme should be used. The modeling of $f_s(\cdot)$ and $g_s(\cdot)$, and their interpolations will be explained at section 4.3.2. Then, next iteration step $n+1$, we should use the sliding yield surface and the elastic limit of displacement for proxy point.

### 4.3.2 Modeling Procedures of Sliding Yield Surface

The sliding yield surface is the elastic limit of the probe tip and the object. In order to detect elastic limit information, we can measure the plastic displacement after a single contact loading and unloading cycle. Thus, whole modeling procedures as follows: 1) Determining the target contact locations. Then, for each contact location, the procedures are, 2) Probing at the target contact point, 3) Loading and unloading in various profiles of the contact force, 4) Finding the border of the elastic contact loading force and the plastic contact loading force from the trend of the plastic displacement, and 5) Making the modeling of the elastic limit of the contact force.

### Modeling Undeformed Surface

The sliding yield surface depends on the initial contact center and its surface normal direction. The surface normal direction should be estimated from the shape of the undeformed surface.

Modeling shape of the undeformed surface is based on the modeling method in chapter 3. However, the detail is slightly different from the chapter 3. One key difference of the system configuration is because the stylus is removed. The probe position is controlled by the computer using PID algorithm. The estimation of the contact point in chapter 3 is also simplified. When the contact is detected, the contact force by PID is unloaded. After unloading is completed, the probe position is used for true contact point for modeling. Moreover, the probe center position is used for modeling to simplify the finding proxy position.

### Selecting Contact Location

Multiple contact locations are required to cover the target area. The current implementation uses a manual section of the contact locations. The user selects the contact locations by moving the probe and contact the real object. Then, the system automatically samples the sliding yield surface for the each contact point.

### Loading-unloading Cycle

Modeling sliding yield surface is based on measuring the plastic displacement for given loading force. The plastic displacement of the probe tip is dependant on the friction properties of the surface and probe tip. This friction property can be affected from contact area, contact pressure, contact velocity, temperature, and so on. However, it is hard to care all possible parameters. In the implementation, the direction and magnitude of the contact force is considered. If the contact force exceed the elastic limit, the probe tip slips off the object or slips to another position on the surface. This slip movement makes the plastic displacement after the unloading is completed.

Our target sliding yield surface is defined as Equation 4.6. The loading force is determined by the magnitude of $f_N$, the direction of $\mathbf{f}_T$, and the magnitude of $\mathbf{f}_T$, because the input variables of $f_\mu$ are $f_N$ and $\mathbf{f}_T$. In order to simplify the problem, eight tangential directions are predefined when the contact location is selected and thus the surface normal is determined. The magnitudes of $f_N$ are also predefined. Then, the magnitude of $\mathbf{f}_T$ is only controlled to find the elastic limit. Thus, we can find the relation between the magnitude of the plastic displacement and the magnitude of the $\mathbf{f}_T$. Moreover, the loading force cannot be reached to the target force instantly. Therefore, we used the force profile in figure 4.6. The time durations of normal loading force and tangential loading force are $t_N$ and $t_T$, respectively. These $t_N$ and $t_T$ values are also used for unloading forces. We fix the rates of $f_N/t_N$ and $f_T/t_T$. The sine function is used to remove the sharp edge of the force profile. One additional parameter is $t_{\text{plateau}}$. It is the duration of applying maximum loading force. This parameter is used to find probe tip slippage. When the other parameters are identical, and only $t_{\text{plateau}}$ is changed, the plastic displacement should not changed if the slip is not occurred. Thus, we fix $f_N$, $\mathbf{f}_N/||\mathbf{f}_N||$, and $\mathbf{f}_T/||\mathbf{f}_T||$ but change $t_{\text{plateau}}$ and $||f_T||$ to find the tendency of plastic displacement.

A single loading-unloading process is as follows. First, $t_{\text{plateau}}$ and $||f_T||$ are determined. Then, the probe moves to the target contact location. When the probe contact with the target surface, the command force is removed completely and the probe tip center is recorded as the initial contact center, $\mathbf{p}_c$. Then, the force profile in figure 4.6 is applied to the target object. When the time is at $t_{\text{measurement}}$, the sampled reaction force $\mathbf{f}_{\text{measurement}}$ and the displacement $\mathbf{u}_{\text{measurement}}$ are captured. After the loading force is completely removed, the release point of the probe tip, $\mathbf{p}_r$ is captured. Then, the plastic displacement $\mathbf{u}^p$ during the loading-unloading process is

$$\mathbf{u}^p = \mathbf{p}_r - \mathbf{p}_c. \tag{4.24}$$

Thus, we have the pair of $\mathbf{u}^p$ and $\mathbf{f}_{\text{measurement}}$.

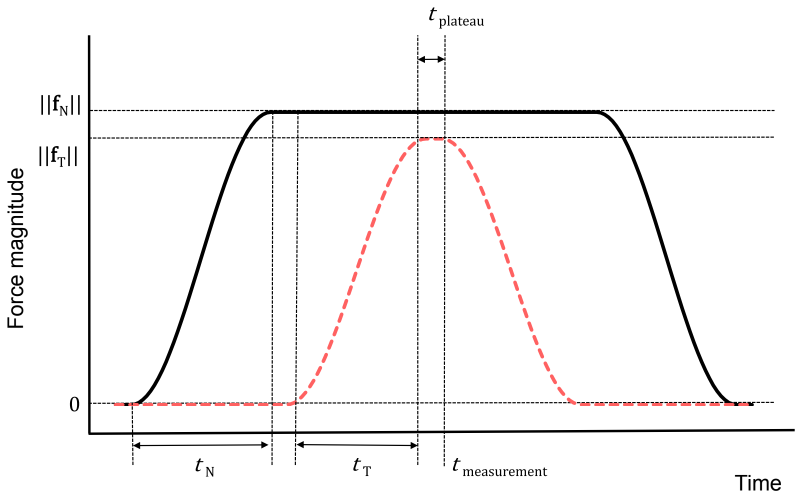


**Fig. 4.6** Profile of the contact loading force. Black solid line is $||\mathbf{f}_N||$ and red dotted line is $||\mathbf{f}_T||$. $t_N$ and $t_T$ are time durations of loading and unloading for normal and tangential forces, respectively. $t_{\text{plateau}}$ determines the duration of applying maximum loading force. $t_{\text{measurement}}$ is measurement time of displacement and reaction force.

### Finding the Elastic Border and Modeling of the Sliding Yield Surface

In order to find the elastic limit, one method is tracking $\mathbf{u}^p$. Ideally, if $\mathbf{u}^p$ has non-zero value, the elastic limit of the contact loading force is already exceed. However, the internal friction of the haptic interface, and the viscosity of the target object consume the energy created by the applying force. These properties make $\mathbf{u}^p$ having non-zero value. Thus, trend between $||\mathbf{u}^p||$ and $f_T$ should be checked to find the slip of the probe tip. If $||\mathbf{u}^p||$ is abruptly changed as increasing of $f_T$, we can consider the point of change as the elastic border.

We also consider an additional information, $t_{\text{plateau}}$. This parameter only controls the time duration of the applying force. If the probe tip and the object in an equilibrium state, $||\mathbf{u}^p||$ is free from changing of $t_{\text{plateau}}$. One drawback of this parameter as a reference for the slippage is that the probe tip slips to another equilibrium position on the surface, $||\mathbf{u}^p||$ is not changed.

We use both tendencies of changing $||\mathbf{u}^p||$ for changing $f_T$ and $t_{\text{plateau}}$. For example, in 4.1, results of measuring the plastic displacement for various $f_T$ and $t_{\text{plateau}}$. $\mathbf{u}^p_{100}$ and $\mathbf{u}^p_{600}$

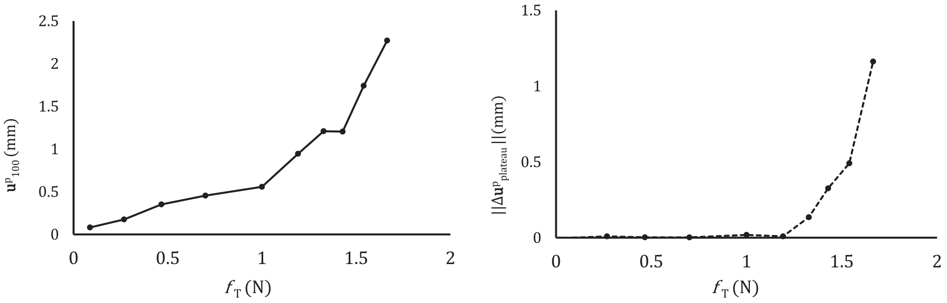


**Fig. 4.7** $f_T$ vs. $||\mathbf{u}^p_{100}||$ at top. $f_T$ vs. $||\Delta\mathbf{u}^p_{\text{plateau}}||$ at bottom.

are plastic displacement when $t_{\text{plateau}}$ are 100 and 600 ms, respectively. $\Delta\mathbf{u}^p_{\text{plateau}}$ is $||\mathbf{u}^p_{600}||$ -$||\mathbf{u}^p_{100}||$

Then, we can find several tendencies as follows. $||\Delta\mathbf{u}^p_{\text{plateau}}||$ is abruptly changed after $f_T$ is over the 1.19 N (See table 4.1 and figure 4.7). Thus, $||\Delta\mathbf{u}^p_{\text{plateau}}||$ should be a good reference to determine the border of elasticity. Figure 4.7 shows that the slope of $||\mathbf{u}^p_{100}||/f_T$ is almost constant before $f_T = 1.5N$. However, even if $||\Delta\mathbf{u}^p_{\text{plateau}}||$ is zero, large $||\mathbf{u}^p_{100}||$ value means that there exist plastic displacement. Thus, we use both $||\Delta\mathbf{u}^p_{\text{plateau}}||$ and $||\mathbf{u}^p_{100}||$. A weighted k-means clustering of two dimensional data is used for finding elastic limit.

Finding the elastic limit surface is as follows. For each contact point, $\mathbf{n}_c$ is given. Then, we have eight predefined tangential directions from $\mathbf{t}_0$ to $\mathbf{t}_7$. We also have two predefined $t_{\text{plateau}}$ values, $t_0$ and $t_1$.

1. Given $f_N$, for each direction of $\mathbf{t}_c$,

   (a) $f_T$ is initialized, and $f_T$ is increased by a predefined step unti If we set the predefined $f_N = \{1,3,5\}$, the modeling time is about one hour for each sliding yield surface. This long modeling time is originated from the rest time to equilibrium. l $f_T$ is reached the predefined limit.

      i. For each $f_T$, $\mathbf{u}^p_{t_0}$ and $\mathbf{u}^p_{t_1}$ are acquired from loading-unloading cycles.

      ii. $||\Delta\mathbf{u}^p_{\text{plateau}}||$ is also estimated for each $f_T$.

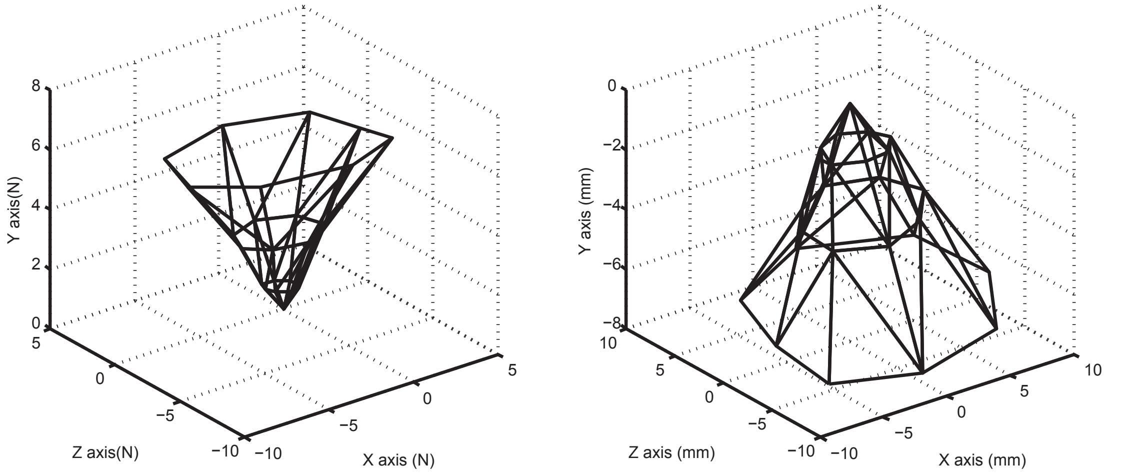
**Table 4.1** Sample plastic displacements for various $f_T$ and $t_{\text{plateau}}$. $f_N$ is 1N. $\mathbf{u}^p_{100}$ and $\mathbf{u}^p_{600}$ are plastic displacement when $t_{\text{plateau}}$ are 100 and 600 ms, respectively. $\Delta\mathbf{u}^p_{\text{plateau}}$ is $||\mathbf{u}^p_{600}||$ -$||\mathbf{u}^p_{100}||$.

| $f_T$ (N) | $\|\|\mathbf{u}^p_{100}\|\|$ (mm) | $\|\|\mathbf{u}^p_{600}\|\|$ (mm) | $\|\|\Delta\mathbf{u}^p_{\text{plateau}}\|\|$ (mm) |
|---|---|---|---|
| 0.09 | 0.087 | 0.08 | 0.00 |
| 0.27 | 0.18 | 0.18 | 0.01 |
| 0.47 | 0.35 | 0.35 | 0.00 |
| 0.70 | 0.45 | 0.46 | 0.00 |
| 1.00 | 0.56 | 0.58 | 0.02 |
| 1.19 | 0.94 | 0.95 | 0.01 |
| 1.33 | 1.21 | 1.34 | 0.14 |
| 1.43 | 1.20 | 1.53 | 0.33 |
| 1.54 | 1.74 | 2.23 | 0.49 |
| 1.66 | 2.27 | 3.43 | 1.16 |

(b) $\mathbf{u}_{\text{elastic limit}}$ and $\mathbf{f}_{\text{elastic limit}}$ are estimated by a weighted k-means clustering of two dimensional data for given $f_N$ and the direction $\mathbf{t}_c$

A set of $\mathbf{u}_{\text{elastic limit}}$ and $\mathbf{f}_{\text{elastic limit}}$ pairs are directly used to determine the sliding yield surface. In order to describe the sliding yield surface, we used a mesh structure. The measured data points are directly used as vertices of the sliding yield surface and elastic limit surface. Then, the other region of the surface is represented as a linear interpolation of the vertices. If we have sufficient vertices, these models can care the non-linear complex data. After selecting the target locations of the sliding yield surface, the modeling procedures are fully automatic. One example of the sliding yield surface and its counter part in the displacement space are shown in Figure 4.8.

For the multiple sliding yield surface, the interpolation among the sliding yield surface is required, because the sliding yield surface depends on the reference point. Thus, if the contact is occurred at the arbitrary point, new sliding yield surface for the arbitrary point is estimated by the interpolation of the preprocessed sliding yield surface. For the interpolation, we make a triangular mesh structure being consisted of the reference points of the preprocessed sliding yield surfaces. Then, the arbitrary contact point is located on the triangular face of the reference points. Using barycentric coordinate system, we can have



(a)Sliding yield surface in force space

(b)Elastic displacement limit surface in displacement space

**Fig. 4.8** Example of the sliding yield surface

the weights for the linear interpolation. The sliding yield surface is represented as a set of the points and its mesh structure. Thus, the interpolation of the sliding yield surface is performed by interpolation of the each vertex point in the sliding yield surface.

## 4.4 Data Acquisition and Modeling

### 4.4.1 Data Acquisition

The data acquisition for training of the interpolation model is applied after the modeling of the sliding yield surface. In the phase of the modeling sliding yield surface, the target object shape and the sliding yield surface models are sampled. Thus, in the current phase, we can estimate a proxy information in the real time interaction.

The basic goal of the data acquisition phase is gathering $\mathbf{p}$, $\dot{\mathbf{p}}$, $\mathbf{q}$, and $\mathbf{f}$. $\mathbf{p}$ and $\mathbf{f}$ are captured by encoder reading and force sensor reading, respectively. $\dot{\mathbf{p}}$ is estimated by FOAW [46]. Then, $\mathbf{q}$ is estimated using the procedures introduced in section 4.3.1 using the sliding yield surface. In the current implementation, we use the elastic displacement limit surface, which is the displacement counter part of the sliding yield surface. The linear interpolation is used to estimate the elastic displacement limit surface for arbitrary reference point. Thus, the modeling area should be inside of the triangular mesh that is consisted of the reference points of the sliding yield surface.

In the data acquisition, the target object is probed with the haptic acquisition system by the human operator. The exploration for data acquisition includes not only pushing but also sliding. Thus, the acquired data covers the area of the target surface. The previous data-driven approaches [32, 89] only used the data explored by pushing. Thus, the modeling data of previous method is depended on the initial contact point and the result force should be interpolated from the point sample. In the previous data-driven methods, the model function should be trained for each initial contact point. The reaction force of the unsampled area is interpolated using the model functions around the contact point. If there are abrupt changes of the object properties in the unsampled area, it cannot be covered. Our method can cover the area explored by the sliding interaction.

### 4.4.2 Modeling

In the data-driven approach, we should approximate a wide range of force responses based on the input data samples. These input data samples should be much smaller than its original response data. For scattered data interpolation, radial basis functions (RBFs) have shown reasonable performance, and used in the previous data-driven haptics [40, 32, 89]. This RBF approximation is a linear superposition of the radial basis functions. We used the computation formulation of the radial basis functions introduced by [89]. With measurement $\mathbf{x}_i$ comprising displacement, proxy, and velocities, and forces $f_i$, the interpolation condition is given as:

$$f_i = \sum_{m=1}^{N} w_m \psi(||\mathbf{x}_i - \mathbf{x}_m||_2), i = 1...N, \tag{4.25}$$

where $N$ is the number of samples, $\psi$ is a polyharmonic spline ( $\psi(r) = r^3$ ), $||\cdot||$ is the Euclidean norm and $w_i$ is a component of the unknown solution vector. Then, it can be written as a linear system of equation: $\mathbf{f} = A\mathbf{w}$, where $\mathbf{f} \in \mathbb{R}^N$ is a force vector and $A \in \mathbb{R}^{N \times N}$ is the interpolation matrix. The RBF reconstruction problem is solved following the random-selection and $l^1$ optimization strategies in Sianov's work [89]. Sianov and Harders' random selection approach is useful for our high dimensional input data set. Our interpolation model includes an additional dimension of the proxy. The finally obtained data-driven model is then used for rendering haptic feedback.

Sianov and Harders introduced an approach inspired by Compressive Sensing and $l^1$-optimization [89]. In our implementation, we used exactly same strategy to train the interpolation function. In the presence of noisy and/or down-sampled data, $l^1$-minimization produces stable and accurate results, thus, we can formulate our RBF reconstruction problem to allow robust recovery of sparse signals. We employ a $l^1$-minimization technique:

$$\min_{\mathbf{w} \in \mathbb{R}^N} \{\lambda ||\mathbf{w}||_1 + \frac{1}{2}||A\mathbf{w} - \mathbf{f}||_2^2\}, \tag{4.26}$$

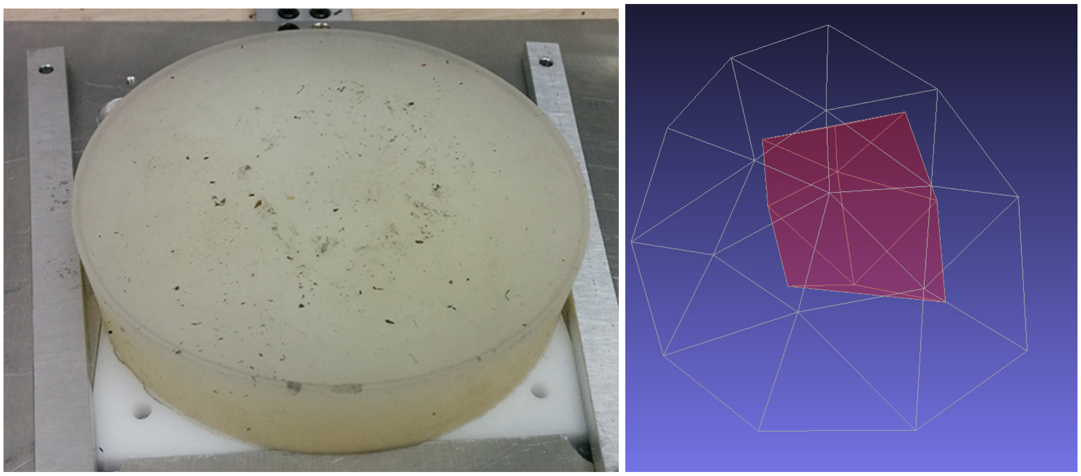
where $||\mathbf{w}||_1$ is a sparsity promoting norm (see e.g. [11], [25]). $\lambda$ is a parameter that controls the trade-off between sparsity and reconstruction fidelity.

Next, we consider methods for reducing the input data, e.g. by compression or selection. Typical RBF modeling approaches are principal component analysis, singular value decomposition, or random projections. While these are possible options, there are memory or computational problems with the involved large matrices with high dimensionality.

Sianov's random selection (RS) strategy is a computationally tractable and resource saving alternative. The approach is inspired by the underlying ideas of Compressive Sensing. We perform a random selection of $M$ recorded measurements $x$ and $f_l$. This random selection is performed by k-means algorithm with $M$ clusters. Based on these we construct the underdetermined RBF matrix $\tilde{A} \in \mathbb{R}^{M \times N}$ and new right hand side $\tilde{f} \in \mathbb{R}^{M}$. For this to be efficient, $M$ should be significantly smaller then $N$: $||w||_0 < M << N$, where $||w||_0$ is number of non-zero components. We use the SpaRSA (Sparse Reconstruction by Separable Approximation) algorithm [99] to solve the problem. The stopping criterion is the number of desired RBFs. For the debiasing we use Conjugate Gradient (tolerance $10^{-9}$ ). With our interpolation model, the whole target area can be covered a single function using RBFs.

### 4.4.3 Clustering

In our interpolation model, the input data set covers large area of the target object. We can train a single interpolation function for the whole input data set. However, if there is large variation of the haptic properties in the area modeled by a single interpolation function, it makes larger deviation. Thus, we spatially separate the target data into several bins by the magnitude of the displacement, $||\mathbf{q} - \mathbf{p}||$ and we model each cell as an interpolation function. The RBF approximation can be considered as the least square fitting and can ignore the small scale error to reduce the large scale error. This clustering can reduce the small scale error in the approximation by collecting similar scaled data.



(a)Appearance (b)Mesh of shape and reference points

**Fig. 4.9** Silicone planar object sample

## 4.5 Performance Evaluation

Our experiments were carried out on a number of different sample deformable objects. The models were created from a mixture of Smooth-On Ecoflex 0030 Silicone Rubber (part A and B) and Silicone Thinner.

### 4.5.1 Inhomogeneous Silicone Planar Object

We mainly present examples for a silicone planar object (See Fig. 4.5.1(a)). It is a wide cylinder of height 2.5 cm and diameter 10 cm. Inside of the object, several pieces of different stiffness silicone are included. These silicone piece contains much more Silicone Thinner. Thus, the stiffness is much lower than the outside cylinder. They change the stiffness of the object, but are not protruded the outside of the shape.

We only modeled the planar top area. Fig. shows the shape mesh of the silicone planar object (white wired mesh), and reference points of the sliding yield surface (red mesh). We sampled nine sliding yield surfaces on the target object. In the sampling of the sliding yield surface, three free defined normal force magnitudes are 1,3,and 5 N.

In the data acquisition phase, about 65,000 samples of $\mathbf{p}$,$\dot{\mathbf{p}}$,$\mathbf{q}$ are captured. Then, the

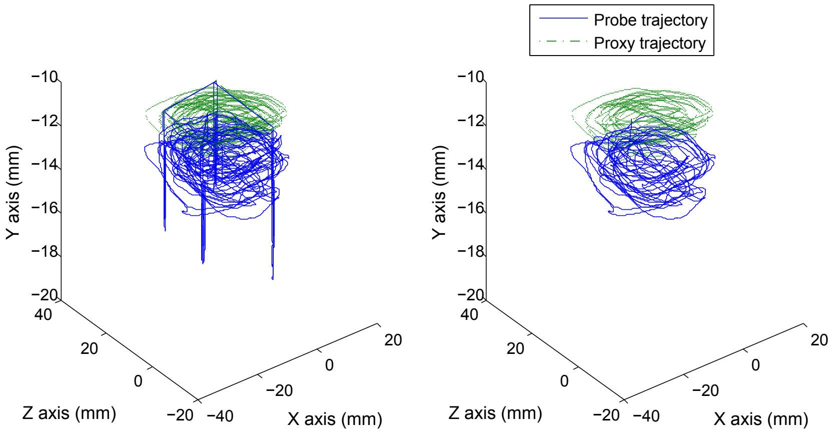


**Fig. 4.10** Sampling trajectory of planar object. About 37,000 samples are tested during about 40 seconds. The left figure is the trajectory of all samples. The right figure shows the trajectory from 20 seconds. The right trajectory contains only sliding interaction.

samples are clustered into five groups by $||\mathbf{q}-\mathbf{p}||$. Each group is modeled as a RBF-functions. In the RBF training phase, $\lambda$ is $0.001||\tilde{A}\tilde{\mathbf{f}}||_\infty$ and the maximum number of the RBF kernel, $M$ is 300.

The constructed data-driven representations were than validated against sensor readings obtained during real interaction. The test trajectories of the probe and proxy represented at Fig. 4.10. About 37,000 samples are tested during about 1 minute. Left image of Fig. 4.10 shows the trajectories of the probe and proxy. This trajectory includes interactions of deep pushing. Right image of Fig. 4.10 shows the only sliding interactions. These trajectories show that our implementation can cover any exploratory interaction in the rendering time. From these trajectories, we can measure and estimate the reaction force.

The trajectory of the force for each axis is shown in Fig. 4.11. The rendered forces are well followed the measured true forces. The trajectory of the force after 20 seconds is shown in Fig. 4.12, which is only including the sliding interaction. These trajectories showed that our algorithm well follows the measured reaction force with any exploration procedures.

We have the measured force $\mathbf{f}_m = \{f_{mx}, f_{my}, f_{mz}\}$ and the estimated commanding force $\tilde{\mathbf{f}}_r = \{\tilde{f}_{rx}, \tilde{f}_{ry}, \tilde{f}_{rz}\}$. We have the absolute force error as

$$\mathrm{AE} = ||\tilde{\mathbf{f}}_r - \mathbf{f}_m||_2. \tag{4.27}$$

Then, we also have the relative force error as

$$\mathrm{RE} = \begin{cases} \dfrac{||\tilde{\mathbf{f}}_r - \mathbf{f}_m||_2}{max(||\tilde{\mathbf{f}}_r||_2, ||\mathbf{f}_m||_2)} & \text{if} ||\mathbf{f}_r - \mathbf{f}_m||_2 > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.28}$$

Although, for general relative error, denominator should be one of $||\tilde{\mathbf{f}}_r||_2$ or $||\mathbf{f}_m||_2$, we use the maximum of two values. Because, one of the forces has zero value, the error should be infinity. In this paper, all the relative force errors should be used as Eq. 4.28. Then, the absolute force error and the relative errors for given axis $i$ are

$$AE_i = |\tilde{f}_{ri} - f_{mi}| \tag{4.29}$$

$$RE_i = \begin{cases} \dfrac{|\tilde{f}_{ri} - f_{mi}|}{max(|\tilde{f}_{ri}|, |f_{mi}|)} & if \quad |\tilde{f}_{ri} - f_{mi}| > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.30}$$

.

Then, the root mean squared error (RMSE) for each axis $i$ is

$$\mathrm{RMSE}_i = \sqrt{\sum_{t=0}^{n} \frac{||f_{mi}(t) - \tilde{f}_{ri}(t)||^2}{n}} \tag{4.31}$$

In the experiment, RMSE of x,y,and z axes are 0.26, 0.15, and 0.25 N, respectively. The median of the force errors of x,y,and z axes are 0.14, 0.085, 0.13 N, respectively. The relative errors are also important for human force perception. The mean and median of relative force error are 0.138 and 0.115, respectively. These force errors are close to the JND of force magnitude (around 10%) [1, 92] and may be noticeable of difference of the measured force. We also show the relative error vs. measured force magnitude graph (See Fig. 4.13). This graph shows that the relative error is noticeable at low measured force area. The distribution of the relative error is also represented as the box plot (See Fig. 4.14). Errors in most of samples are lower than 20%.

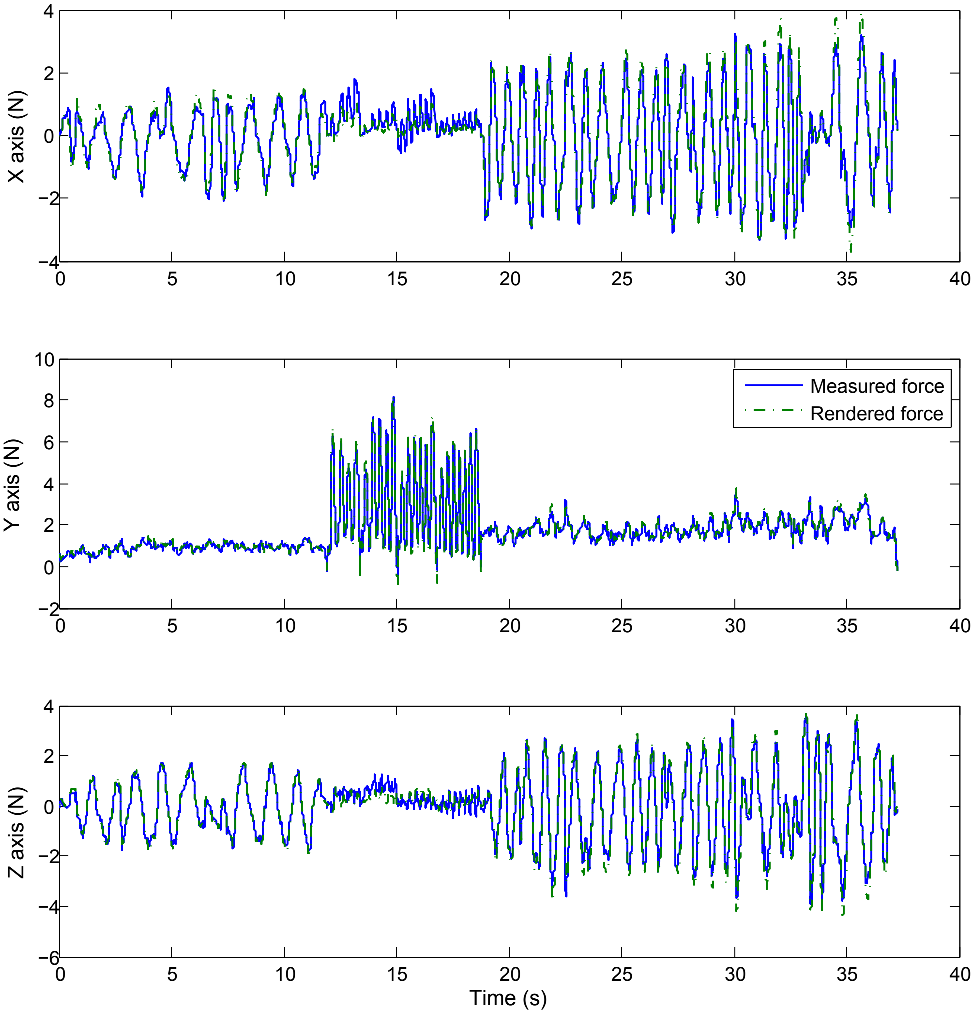


**Fig. 4.11** The trajectory of the force in planar silicone object

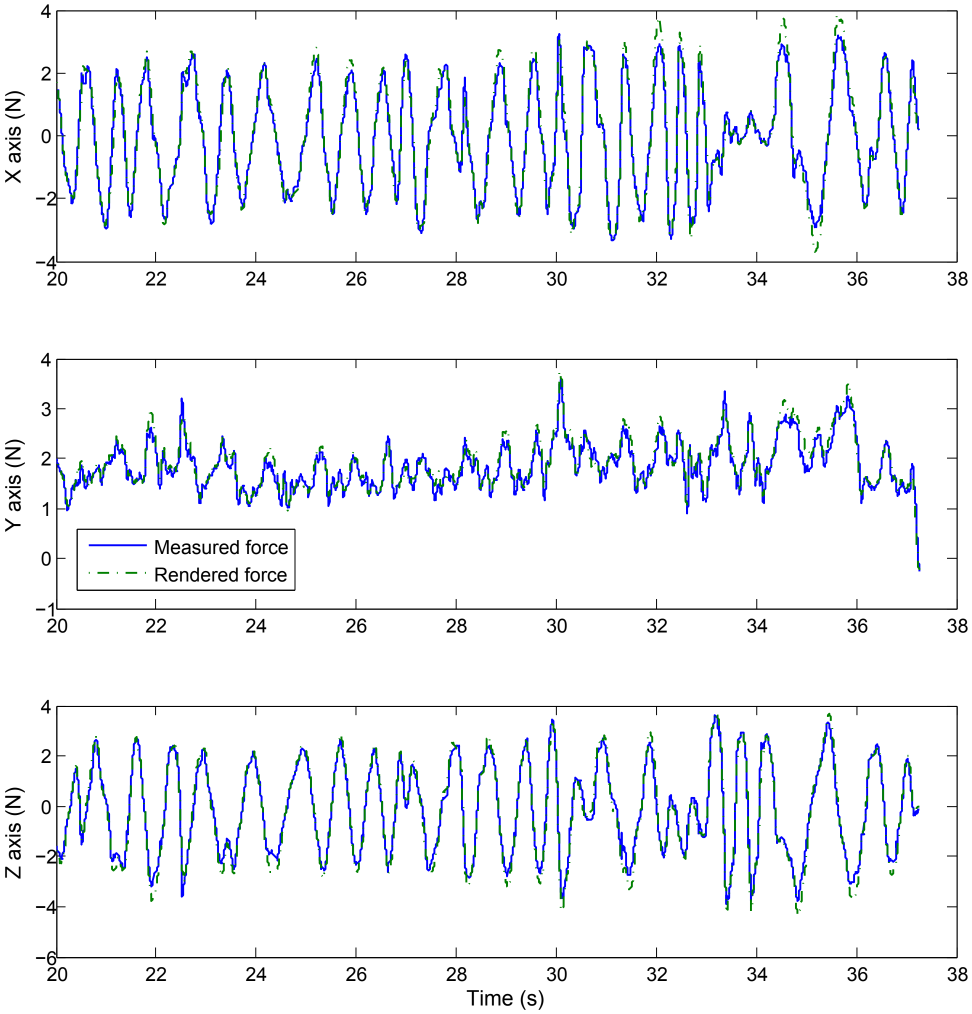


**Fig. 4.12** The trajectory of the force in planar silicone object for sliding interaction

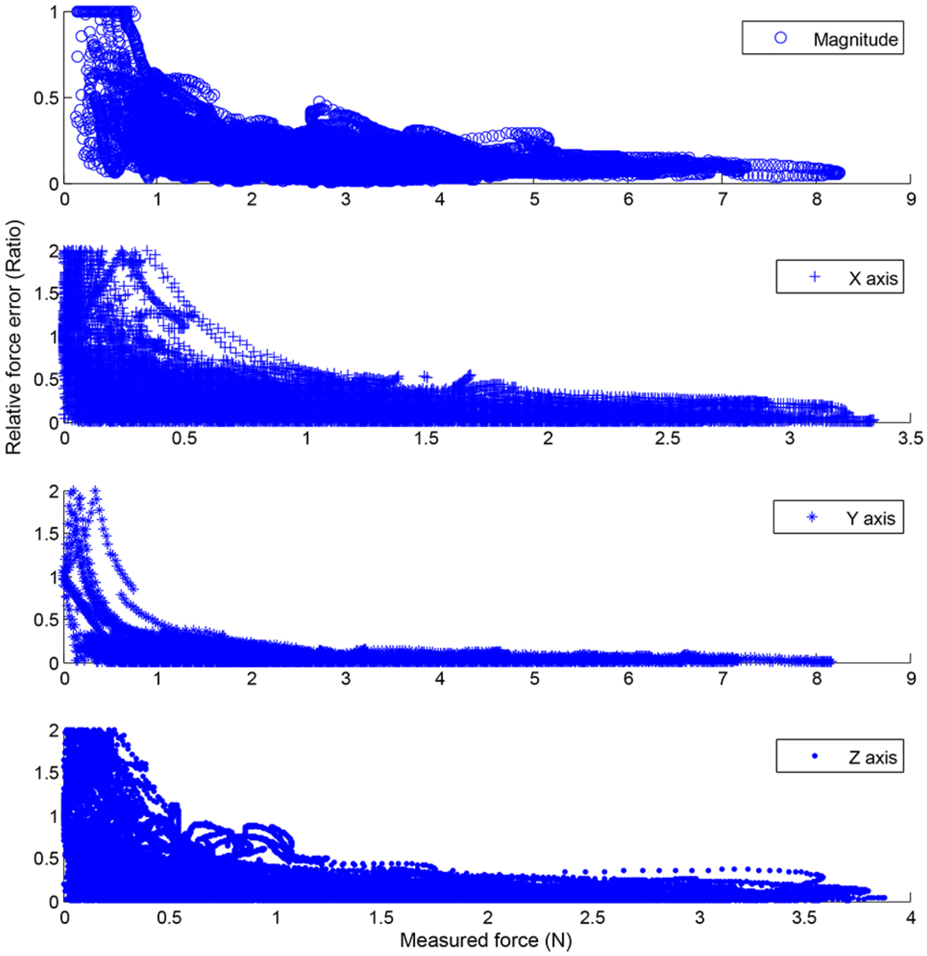


**Fig. 4.13** Relative error of the force magnitude along the test trajectory for the silicone planar object

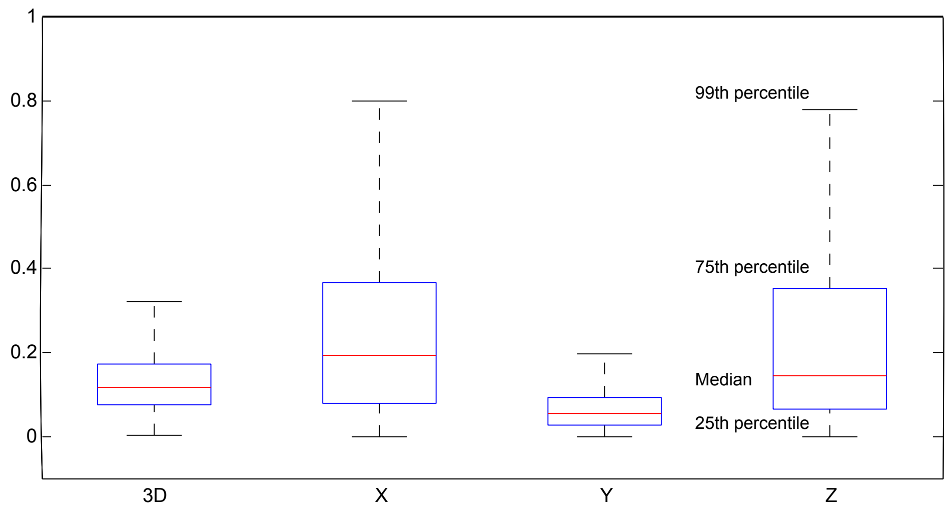
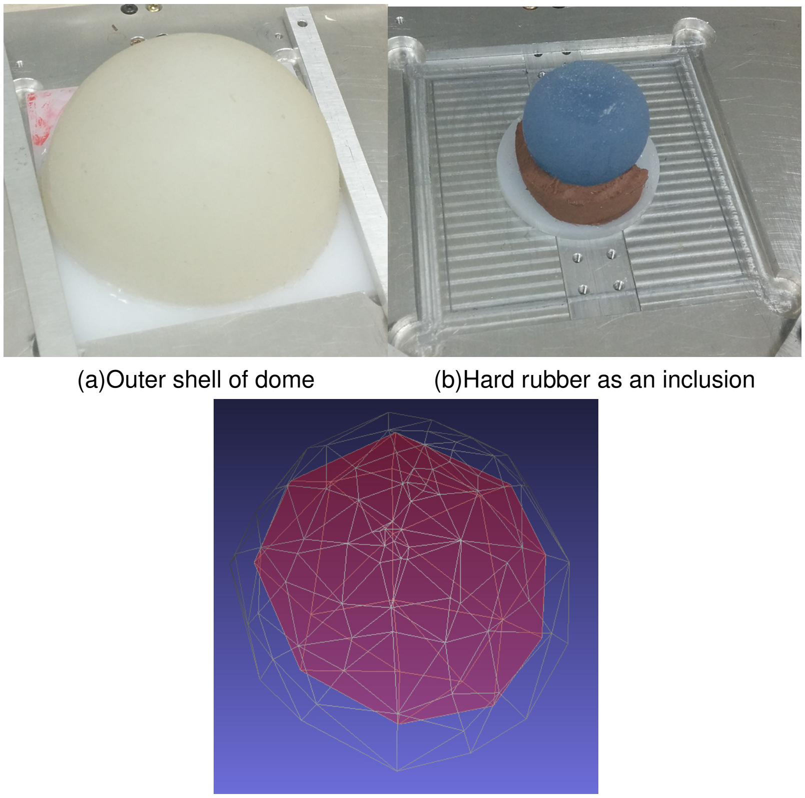


**Fig. 4.14** Box plot of the relative force error of the planar silicone. 3D shows the relative error by Eq. 4.28 . X,Y,and Z showed the relative error for the axes by Eq. 4.30



(a)Outer shell of dome

(b)Hard rubber as an inclusion

(c)Mesh of shape and reference points

**Fig. 4.15** Silicone dome

### 4.5.2 Silicone Rubber Dome with Inclusion

The another example is a silicone dome (diameter 10cm) (See Fig. 4.5.2). Inside the silicone outer cell, a stiff rubber ball is located. Thus, at the top and center of the dome, the stiffness is higher than surround. Fig. 4.5.2 shows the shape mesh of the silicone dome (white wired mesh), and reference points of the sliding yield surface (red mesh). We sampled 18 sliding yield surfaces on the target object. In the sampling of the sliding yield surface, three free defined normal force magnitudes are 1,3,and 5 N. In the data acquisition phase, about 52,000 samples of $\mathbf{p}$,$\dot{\mathbf{p}}$,$\mathbf{q}$ are captured. Then, the samples are clustered into five groups by $||\mathbf{q}-\mathbf{p}||$. Each group is modeled as a RBF-functions. In the RBF training

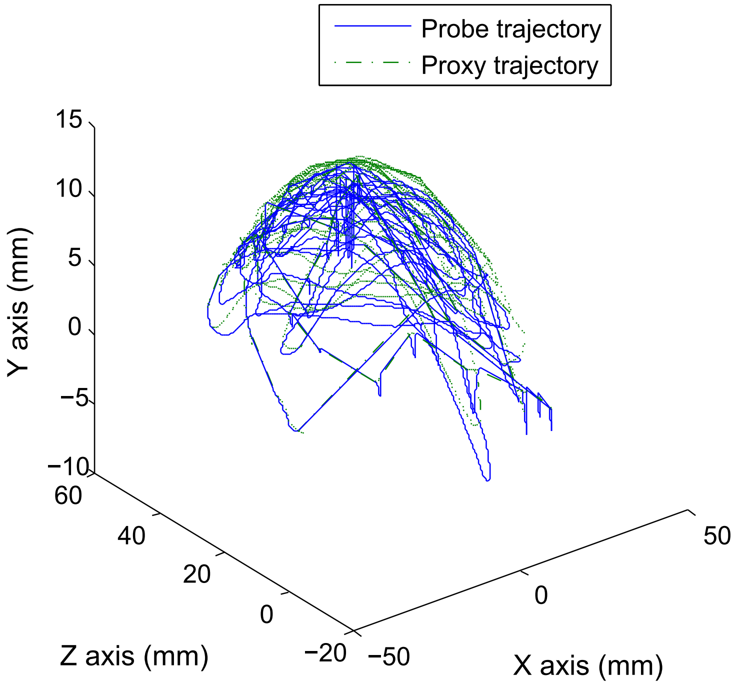


**Fig. 4.16** Sampling trajectory in silicone dome. About 24,000 samples are tested during about 30 seconds.

phase, $\lambda$ is $0.001||\tilde{A}\tilde{\mathbf{f}}||_\infty$ and the maximum number of the RBF kernel, $M$ is 300.

The constructed data-driven representations were than validated against sensor readings obtained during real interaction. The test trajectories of the probe and proxy represented at Fig. 4.16. About 24,000 samples are tested during about 30 seconds. The trajectory shows that our implementation can cover any exploratory interaction with non-planar object. From these trajectories, we can measure and estimate the reaction force.

The trajectory of the force for each axis is shown in Fig. 4.17. The rendered forces are well followed the measured true forces. These trajectories showed that our algorithm well follows the measured reaction force with any exploration procedures. Moreover, the rendering algorithm is well operated on the non-planar surface.

In the experiment with silicone dome, RMSE of x,y,and z axes are 0.51, 0.33, and 0.50 N, respectively. The median of the force errors of x,y,and z axes are 0.27, 0.17, 0.29 N, respectively. The mean and median of relative force error are 0.260 and 0.211, respectively. We also show the relative error vs. measured force magnitude graph (See Fig. 4.18). This graph shows that the relative error is noticeable at low measured force area. The distribution of the relative error is also represented as the box plot (See Fig. 4.19).

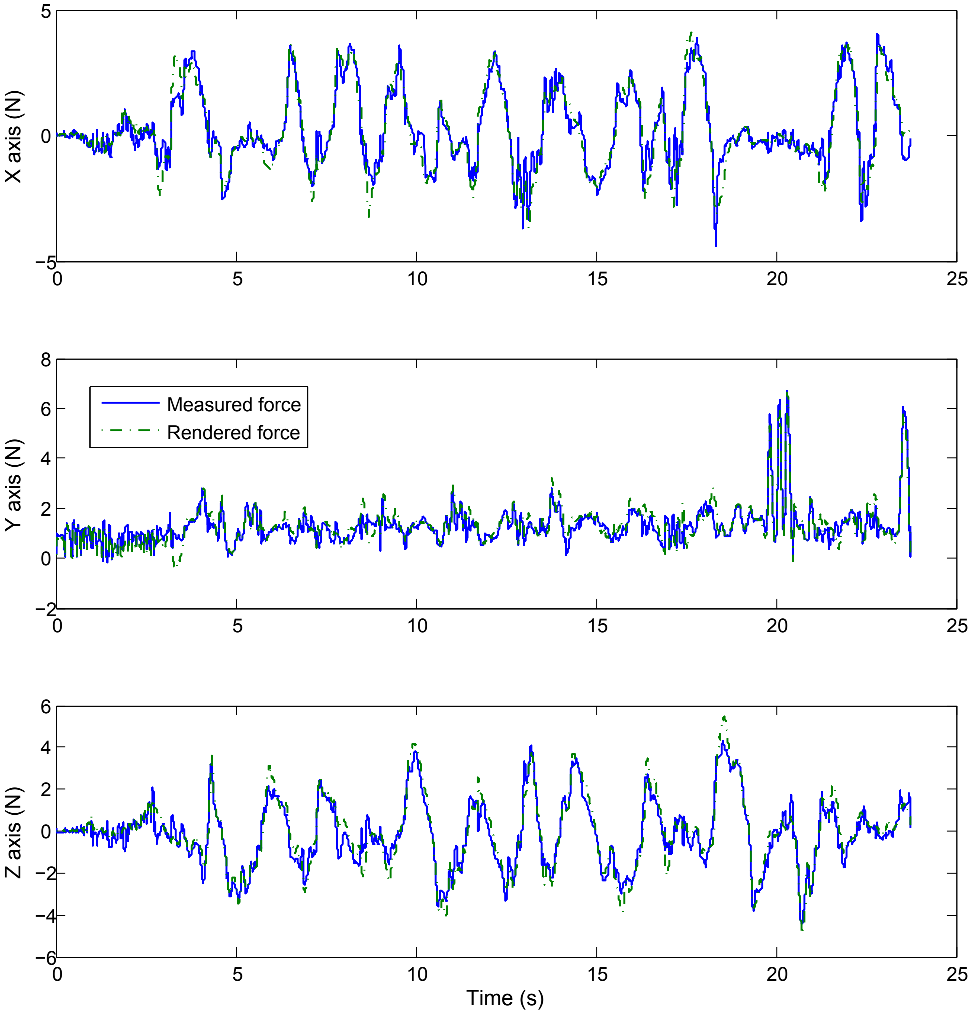


**Fig. 4.17** The trajectory of the force in silicone dome.

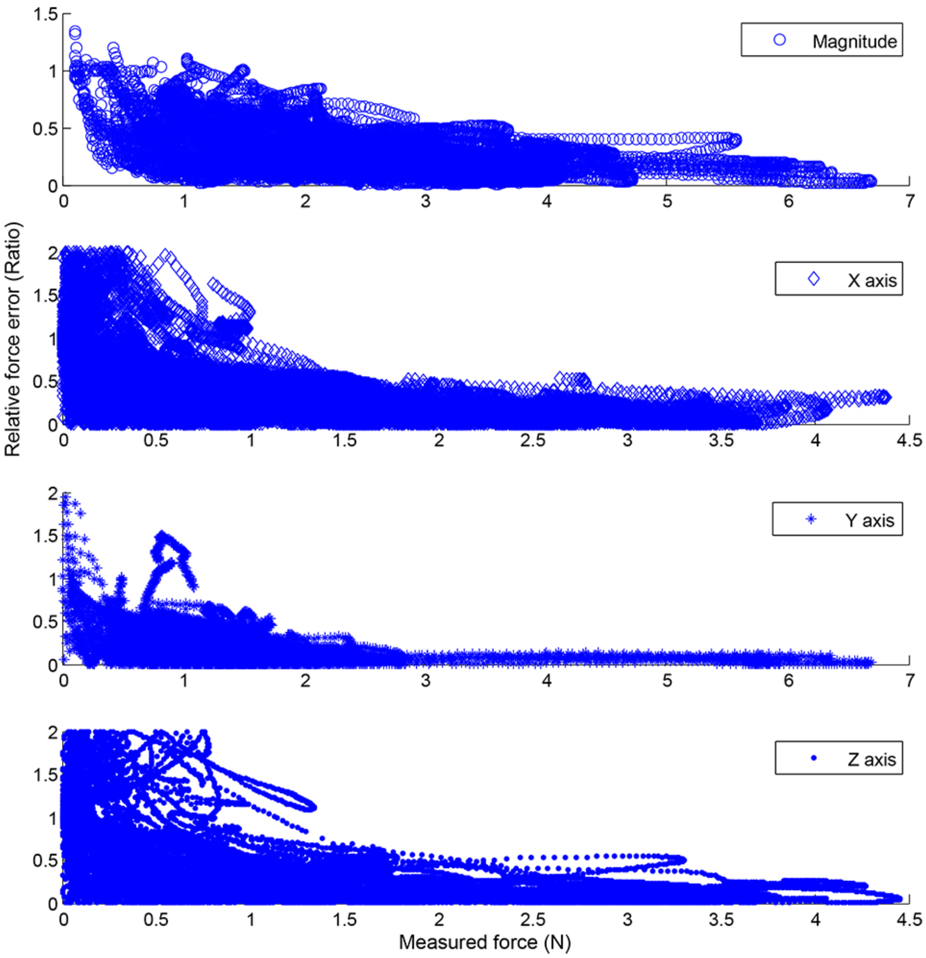


**Fig. 4.18** Relative error of the force magnitude along the test trajectory for the silicone dome

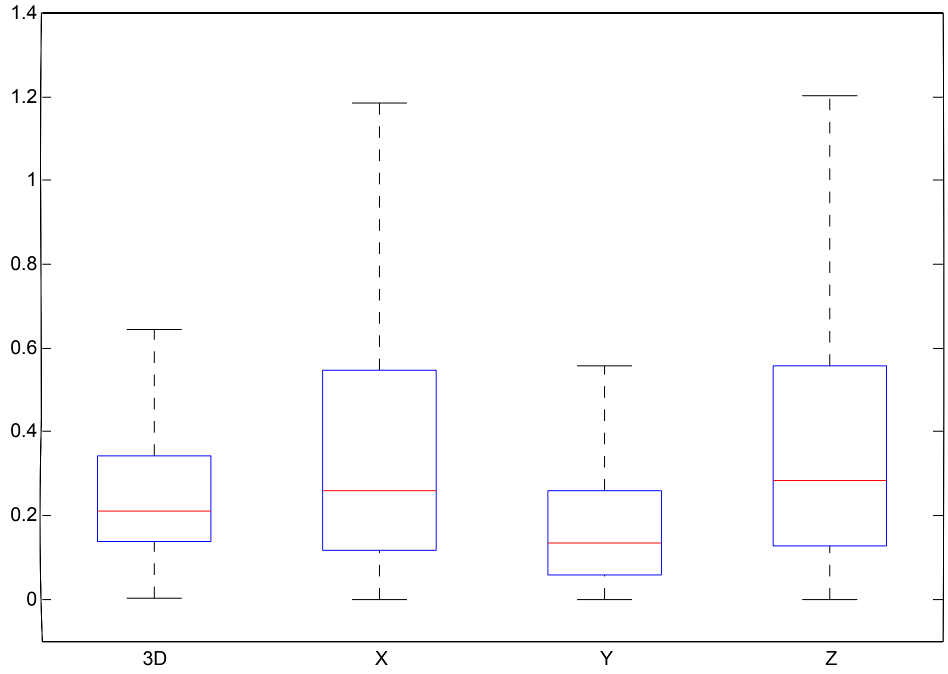


**Fig. 4.19** Box plot of the relative force error of the planar silicone. 3D shows the relative error by Eq. 4.28 . X,Y,and Z showed the relative error for the axes by Eq. 4.30

## 4.6 General Discussion

Our modeling system can model the inhomogeneous soft-deformable object. Especially, allowing to user the data-acquisition in the unconstrained exploration is unique aspect of our modeling framework. In the performance evaluation, we prove the ability of the modeling the inhomogeneous objects with simple planar and non-planar shapes. Especially, our modeling system does not limit the exploratory procedure in not only data acquisition but also rendering time. These properties are useful to handle the real inhomogeneous objects. Especially, for the haptic augmented reality, our data-driven model can be used to change the haptic properties.

The drawback of our work in the performance evaluation is the modeling fidelity is lower than previous researches. The relative error is higher than human force magnitude threshold (about 10%). This error can be originated from the error in estimating the proxy position. The frictional phenomenon is very complicate. However, we use a relatively simple measurement based model to handle the frictional phenomenon. The more complicate model to describe the friction with the elasto-plasticity should be required [36, 75]. The second major error source is the estimating velocity. We use a FOAW filter to estimate the velocity. However, the accuracy of the velocity has a close relationship with the encoder accuracy. In the previous researches, the encoders of PHANToM are upgraded [89, 38]. We can consider the upgrading the measurement systems. Finally, strong virtual fixture also can be improve the accuracy by controlling human error [89].

Chapter 5

# Data peceptualization of Inhomogeneous Haptic Data: Case Study on Transferring of Shape and Stiffness

In this chapter, we report the data-perceptualization application to consider the inhomogeneity of the target data. The proposed application resolves the perceptual distortion of the shape and stiffness perception. This is a representative case of the modeling and transfer of haptic data with a inhomogeneity. In *data perceptualization*, the properties of a dataset are conveyed to the user through multi-modal sensory channels including vision (visualization), sound (sonification), and touch (haptization). Data perceptualization enables the user to see, hear, and/or touch the data with an increased bandwidth of information transmission [8]. One of the components central to effective data perceptualization is a *transfer function*, which maps a data variable (e.g., the density of a voxel) to a display attribute (e.g., color, pitch, or force). A transfer function must guarantee that the information *perceived* by the user matches the original information contained in the dataset. Otherwise, perceptualization can give the user an incorrect understanding of the data properties.

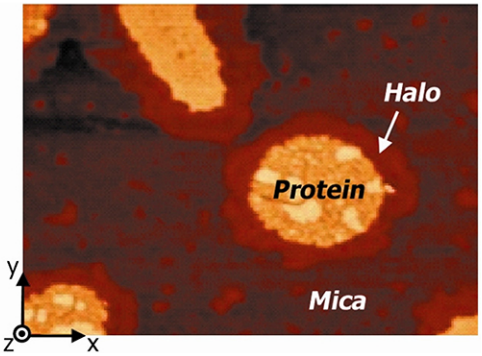


**Fig. 5.1** The surface height map of "protein-on-mica" data. Higher regions are coded with brighter colors.

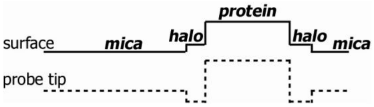


**Fig. 5.2** A cross-section of the height map shown in Fig. 5.1 along with the typical trajectory of a haptic probe tip.

## 5.1 Data-perceptualization of Shape and Inhomogeneous Stiffness Data

Our work is motivated from an example is provided in Fig. 5.1, which was originally presented in [18]. This figure shows a surface height map of bi-lipid membrane patches with embedded proteins on a mica substrate, with brighter colors corresponding to higher regions. This topography map was acquired using an atomic force microscope, which could measure various collocated nano-scale features on a material surface, including the topography and stiffness distribution [87]. The given image shows the protein membrane patches surrounded by a halo (presumably of the lipids that have dissociated from the membrane) resting on the atomically flat mica substrate. The membrane patch is filled with a periodic array of transmembrane protein, so it is considerably stiffer than the halo of the dissociated lipids, but not as stiff as the mica substrate. The surface height profile of this dataset is depicted in 5.2 by a solid line. For the sake of convenience, this dataset is called "protein-on-mica" in this paper.

In [96, 18], we reported that the protein-on-mica dataset rendered with the traditional penalty-based algorithm can impart a distorted perception of its height information. In opposite to the height model, users often felt that the halo regions were lower than the surrounding mica region. This observation was also confirmed in the probe trajectories recorded during a user's strokes on the virtual surface. A typical probe position trace simplified for illustration is shown in Fig. 5.2 with a dotted line, which indicates that the probe was "dipped" into the halo region against the height profile.

To identify the underlying reasons for this, we proposed and experimentally validated a *force constancy theory*, which states that humans tend to maintain a constant contact force while scanning a surface in order to perceive its shape [96, 18]. This theory can explain why the halo region with lower stiffness in Fig. 5.2 allows larger penetration than the mica with higher stiffness. If the increase of penetration in the halo region due to the lower stiffness exceeds the height difference between the halo and the mica regions, the halo region is perceived to be lower than the mica region. This phenomenon of force constancy was also confirmed by or utilized in other studies [14, 17, 15, 33, 16, 95, 47, 48, 79]. According to the well-established perceptual theory of exploratory procedures (EPs) [60], the user taps on a surface to perceive its stiffness for optimal perceptual performance, while s/he strokes the surface to feel its shape. Combining the two theories leads to the possibility of an incorrect haptization of the surface topography with a continuously varying stiffness distribution.

Therefore, we set the research goal as perceptually accurate haptization of a surface topography with spatially varying stiffness. We previously developed a series of *Topography Compensation Algorithms (TCAs)* [14, 15, 16] based on the force constancy theory. These algorithms estimate a user-applied force and adjust the height profile to compensate for the height distortion. They were designed for the height and stiffness data represented by *images* and demonstrated perceptually accurate haptization of the surface topography.

In this chapter, we present an algorithm (TCA-MESH) to haptize an object shape with a continuous stiffness distribution when the data are represented by a *mesh*, a considerably more general data structure. We assign a stiffness value to each vertex of the mesh in order to represent a stiffness distribution over the mesh surface. The previous TCAs for the image-represented data are extended to take care of meshes, with the following two primary improvements. First, the entire algorithm is adapted to work with a constraint-based haptic rendering approach and an existing collision detection algorithm for meshes. To this end, new strategies including the translation of meshes as well as many implementation techniques for mesh management are employed. Second, a new method is developed for estimating the user-applied force to the mesh surface, where a large number of sharp normal direction changes may exist.

The performance of TCA-MESH is demonstrated by means of the probe trajectories collected from complex objects. Its perceptual performance is also examined via a user study for both shape and stiffness haptization.

## 5.2 Topography Compensation Algorithm for Mesh

### 5.2.1 Algorithm Overview

TCA-MESH uses a mesh $M$ that stores the shape and stiffness of perceptualization data. Each vertex of $M$ is assigned with a normal vector and a stiffness value. For a point $\mathbf{q}$ on the surface of $M$, we can obtain its normal $\mathbf{n}(\mathbf{q})$ and stiffness $k(\mathbf{q})$ using barycentric interpolation. Three examples of such meshes are shown in Fig. 5.3.

Let $n$ be the time index for haptic rendering loops. Important points and meshes used in TCA-MESH and their relationships are depicted in Fig. 5.4. The probe position of a haptic interface, called the haptic interface point (HIP), is denoted by $\mathbf{p}(n)$. If contact occurs with $\mathbf{p}(n)$, the ideal haptic interface point (IHIP) (also called the god object or virtual proxy), $\mathbf{q}(n)$, that is constrained on the mesh $M$ can be determined using a general constraint-based algorithm [102, 84, 37]. Then, the response force can be computed using $k(\mathbf{q})$, $\mathbf{n}(\mathbf{q})$, and the penetration depth. This simple method is effective in rendering the stiffness of a virtual object as well as its geometry, even for non-constant stiffness. The range of the stiffness that allows stable rendering depends on the haptic interface (virtual coupling with zero damping). We will call this method the constraint-based algorithm (CBA). CBA is subject to the aforementioned topography compensation problem.

TCA-MESH considers a *compensated mesh* $M^{\mathbf{t}(n)}$ in which the vertices are translated by

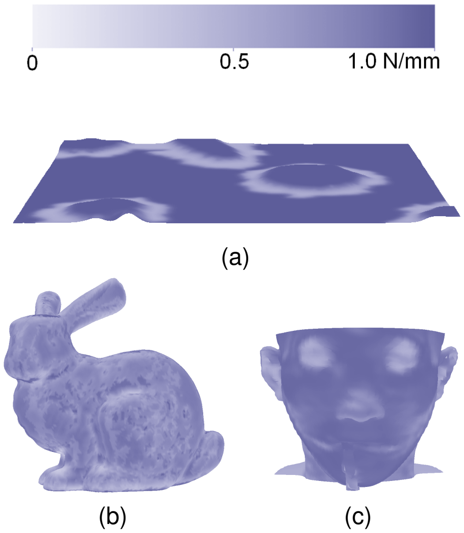


**Fig. 5.3** Mesh models with a continuous stiffness distribution. Higher stiffness values are represented by darker colors. The stiffness range of all the meshes is 0–1.0 N/mm. (a) The protein-on-mica model, (b) the Stanford bunny model [90], and (c) a human head model.

a vector $\mathbf{t}(n)$ from the original vertices of $M$. IHIP $\mathbf{q}(n)$ is constrained on $M^{\mathbf{t}(n)}$, not on $M$. From $\mathbf{q}(n)$, we find a *compensated ideal haptic interface point (CIHIP)*, $\mathbf{q}'(n)$:

$$\mathbf{q}'(n) = \mathbf{q}(n) + \mathbf{s}(n), \tag{5.1}$$

The output force $\mathbf{f}(n)$ is computed using $\mathbf{q}'(n)$ and $\mathbf{p}(n)$.

Here, $\mathbf{s}(n)$, called the point compensation vector, represents the translation of IHIP necessary for accurate shape rendering in spite of stiffness changes. $\mathbf{t}(n)$, the mesh compensation vector, preserves the accumulated effect of $\mathbf{s}(n)$ by translating the entire mesh to enable seamless interaction between successive contacts. The computation of these two terms is the core of TCA-MESH, and it uses the force applied by the user onto a virtual surface that is estimated on the basis of the force constancy theory.

The overall behaviors of TCA-MESH are described below (also see Fig. 5.5).

- S1 (Fig. 5.5a): The algorithm initializes $\mathbf{t}(0) = \mathbf{0}$ and $\mathbf{s}(0) = \mathbf{0}$.
- S2 (Figs. 5.5b and 5.5c): After the first contact, the algorithm updates $\mathbf{s}(n)$ in a way that minimizes the shape distortion. At same time, $\mathbf{s}(n)$ is restored to $\mathbf{0}$, gradually

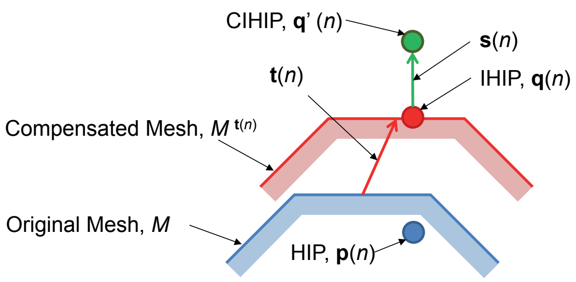


**Fig. 5.4** Meshes and major points used in TCA-MESH.

and imperceptibly, to prevent the effect of inevitable errors on the estimation of the user-applied force. $\mathbf{t}(n)$ remains as $\mathbf{0}$.

- S3 (Fig. 5.5d): When the contact is released, the surface offset due to $\mathbf{s}(n)$ should be taken into account for subsequent contact detections. Otherwise, the user can easily notice an abrupt change in the surface position. To this end, $\mathbf{t}(n+1)$ is set to $\mathbf{s}(n)$, and $\mathbf{s}(n)$ is reset to $\mathbf{0}$. The translated mesh $M^{\mathbf{t}(n)}$ is used for subsequent collision detections. While no contact occurs, $\mathbf{t}(n)$ is also iteratively restored to $\mathbf{0}$ to prevent the accumulation of the mesh offsets.

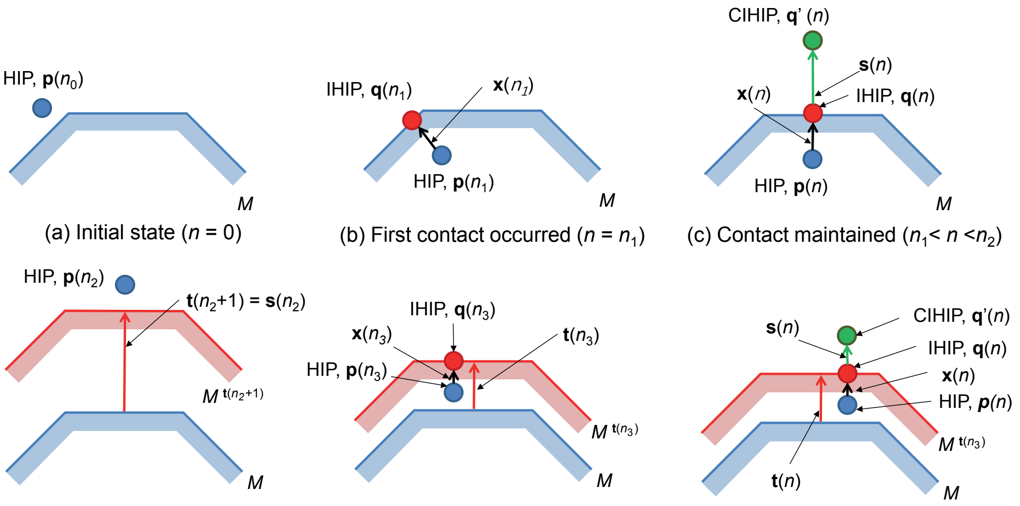
- S4 (Fig. 5.5e): When another contact is detected, $\mathbf{t}(n)$ stops being updated, and the algorithm starts calculating $\mathbf{s}(n)$ based on $M^{\mathbf{t}(n)}$. At this moment, $\mathbf{t}(n)$ may or not be $\mathbf{0}$. In the former case, the translated mesh has been fully restored to the original mesh. The latter case can occur when the user touches the surface quickly and repeatedly; maintaining the translated mesh enables the user not to notice the surface offset.

- S5 (Fig. 5.5f): The algorithm is repeated from S2, except that $\mathbf{t}(n)$ may no longer be $\mathbf{0}$.

Further details are presented below.

### 5.2.2 Initial State

At the beginning of rendering ($n = 0$; Fig. 5.5a), TCA-MESH assumes no collision between the HIP $\mathbf{p}(0)$ and the mesh $M$, rendering zero force. Several variables are initialized as



(a) Initial state ($n$ = 0)

(b) First contact occurred ($n = n_1$)

(c) Contact maintained ($n_1 < n < n_2$)

(d) Contact released ($n = n_2$)

(e) Another contact occurred ($n = n_3$)

(f) Contact maintained ($n > n_3$)

**Fig. 5.5: Behaviors of TCA-MESH.**

follows: $\mathbf{s}(0) \leftarrow \mathbf{0}$, $s(0) \leftarrow 0$, $\mathbf{t}(0) \leftarrow \mathbf{0}$, and COLLISION←FALSE, where $s(n)$ denotes the signed magnitude of $\mathbf{s}(n)$. This state is maintained until the first contact occurs between the line segment from $\mathbf{p}(n-1)$ to $\mathbf{p}(n)$ (representing the probe trajectory) and the mesh $M$. Then, COLLISION is set to TRUE, and TCA-MESH transitions to the next state. Contact detection between a line and a mesh is supported in most collision detection packages, and our implementation uses the OBB tree [34].

### 5.2.3 When First Contact Occurs

Assume that the first contact occurs at $n_1$ (Fig. 5.5b), that is, $\mathbf{p}(n_1)$ has entered $M$. Then, we find IHIP $\mathbf{q}(n_1)$ that is constrained on the surface of $M$ by using an efficient point-based 3-DOF rendering algorithm [37]. We also perform a force shading algorithm for smooth movements of $\mathbf{q}(n)$ around edges [70].

The stiffness value to render in this frame is $k(n_1)$ $(= k(\mathbf{q}(n_1))$, for the sake of simplicity). Compensation for shape distortion is not necessary yet because HIP has just initiated contact. Thus, $\mathbf{t}(n_1) = \mathbf{0}$, $\mathbf{s}(n_1) = \mathbf{0}$, $s(n_1) = 0$, and CIHIP $\mathbf{q}'(n) = \mathbf{q}(n)$. Then the response force is

$$\mathbf{f}(n_1) = k(n_1)\,(\mathbf{q}(n_1) - \mathbf{p}(n_1)). \tag{5.2}$$

The algorithms in Section 5.2.2 and 5.2.3 are equivalent to the typical 3-DOF constraint-based algorithm.

### 5.2.4 While Contact is Maintained

In later frames ($n_1 < n < n_2$), we assume that the contact made at $n_1$ is maintained until it is released at $n_2$ (Fig. 5.5c). In this state, $M^{\mathbf{t}(n)}$ is still the same as $M$; $\mathbf{t}(n)$ is updated when the contact is released (Section 5.2.5). We find IHIP $\mathbf{q}(n)$ on $M$ using the efficient point-based algorithm [37].

One of the key components of TCA-MESH is the determination of the point compensation vector $\mathbf{s}(n)$. CIHIP $\mathbf{q}'(n)$ is computed by (5.1). This $\mathbf{q}'(n)$ is used for computing a response force that removes the height distortion error. Here, $\mathbf{s}(n) = s(n)\mathbf{n}(n)$, where $\mathbf{n}(n)$ is the outward normal of the polygon on which $\mathbf{q}(n)$ stays. Thus, the problem of finding $\mathbf{s}(n)$ reduces to that of finding $s(n)$. $s(n)$ represents the amount of height compensation necessary to remove the height distortion caused by non-uniform stiffness. We compute $s(n)$ recursively such that:

$$s(n) = s(n-1) + \Delta s(n), \tag{5.3}$$

$$\Delta s(n) = \frac{k(n-1) - k(n)}{k(n)}\,(x(n) + s(n-1)), \tag{5.4}$$

where $s(0) = 0$. $x(n)$ denotes the signed penetration depth, that is, the signed distance from $\mathbf{p}(n)$ to $\mathbf{q}(n)$. In (5.4), the product of $k(n-1) - k(n)$ and $x(n) + s(n-1)$ is the increase in force that would be rendered due to the stiffness change if no compensation were to be performed in this frame. This force increase is divided by the current stiffness $k(n)$, yielding the desired displacement change, $\Delta s(n)$, for appropriate height compensation.

Then, the response force is computed by

$$\mathbf{f}(n) = k(n)\,(\mathbf{q}'(n) - \mathbf{p}(n)) = k(n)(x(n) + s(n))\,\mathbf{n}(n). \tag{5.5}$$

The correctness of this algorithm is proven in Lemma 1.

**Lemma 1** *Assuming that the user maintains a constant force and that the haptic interface provides perfect transparency (infinite-gain force tracking), then $\Delta s(n) = -\Delta h_d(n)$, where $\Delta h_d(n)$ is the amount of height distortion caused by non-uniform stiffness.*

**Proof** This lemma is proven using mathematical induction. When the first contact is made at $n = n_1$, $h_d(n_1) = 0$; therefore, there exists no need for height compensation. Next, assume that $h_d(n-1) = -s(n-1)$, i.e., perfect height compensation at $n-1$. Then, we need to show that $h_d(n) = -s(n)$, which is equivalent to $\Delta h_d(n) = h_d(n) - h_d(n-1) = -\Delta s(n)$. If the user maintains a constant force between $n-1$ and $n$, $\Delta h_d(n)$ is as follows:

$$\Delta h_d(n) = \begin{cases} 0 \text{if } k(n) = k(n-1) \\ -f_p(n)\left(\frac{1}{k(n)} - \frac{1}{k(n-1)}\right) \text{if } k(n) \neq k(n-1) \end{cases}, \tag{5.6}$$

where $f_p(n)$ is the magnitude of a user-applied force [18].

The assumption of force constancy means that the user-applied force and the output force of the haptic interface are the same. The assumption of perfect transparency also allows the output force to be the same as the command force to the haptic interface. Hence, the user-applied force is equivalent to the force command:

$$f_p(n) = |\mathbf{f}(n)|_s = k(n)(x(n) + s(n)), \tag{5.7}$$

where $|\cdot|_s$ represents the signed magnitude.

Then $\Delta h_d(n)$ in (5.6) is

$$\Delta h_d(n) = -f_p(n)\left(\frac{1}{k(n)} - \frac{1}{k(n-1)}\right) \tag{5.8}$$

$$= -k(n)\left(x(n) + s(n)\right)\left(\frac{1}{k(n)} - \frac{1}{k(n-1)}\right). \tag{5.9}$$

Using (5.3) and (5.4), it is straightforward to show that

$$\Delta h_d(n) = -\frac{k(n-1) - k(n)}{k(n)}\left(x(n) + s(n-1)\right) = -\Delta s(n). \tag{5.10}$$

This completes the proof.

In the proposed algorithm, the estimation of the user-applied force $f_p(n)$ is the key to accurate height compensation. The previous TCAs [14, 15, 16] used a rather simple estimator: $f_p(n) = f(n-1)$. This was based on an approximation that the force output at $n$ is similar to the sample earlier force command at $n-1$. This estimator works well when the directional changes of geometry are smooth. However, we have found that it can cause unacceptable errors with mesh models, which can include a large number of sharp directional differences around edges. The normal component of the user-applied force determines the penetration depth, which subsequently determines the response force. On flat or smooth surfaces, their normal direction changes slowly, so the force projected to the surface normal also varies slowly if the user-applied force is constant. This is the necessary condition for the previous TCAs to provide adequate height compensation. However, with mesh models, the surface normal can vary abruptly around edges, and so does the projected normal component of the user-applied force even if the user-applied force is constant. This makes the user-applied force estimator of the previous TCAs, $f_p(n) = f(n-1)$, contaminated with a large error. Therefore, we use $f(n)$, the force magnitude that will be rendered in the current frame $n$, as the force estimator instead of $f(n-1)$.

Finally, $f(n)$ is determined from the current compensated penetration depth, $x(n)+s(n)$. This rule can improve the quality of rendering to a large extent for mesh models.

The above algorithm, however, is contingent upon the assumptions that the user applies a strictly constant force and the haptic interface is perfectly transparent, either of which cannot be true in reality. Thus, it is necessary to remove the inevitable errors of height compensation that can be accumulated over time. To this end, TCA-MESH uses a height restoration term $\Delta r_s(n)$, similarly to the previous TCA [16], such that

$$\Delta r_s(n) = \begin{cases} -a_s d_s(n) \text{if } s(n-1) > a_s d_s(n) \\ -s(n-1) \text{if } |s(n-1)| \leq a_s d_s(n) \\ a_s d_s(n) \text{if } s(n-1) < -a_s d_s(n) \end{cases}, \tag{5.11}$$

where $a_s$ is the slope for height restoration and $d_s(n)$ represents the lateral displacement of the probe. $d_s(n)$ is determined to be the projection of the probe trajectory, $\mathbf{p}(n) - \mathbf{p}(n-1)$, onto the tangent plane of $\mathbf{n}(n)$:

$$d_s(n) = ||(\mathbf{p}(n) - \mathbf{p}(n-1)) \times \mathbf{n}(n)||. \tag{5.12}$$

The two terms, compensation term $\Delta s(n)$ and restoration term $\Delta r_s(n)$, compete with each other. While $\Delta s(n)$ makes CIHIP deviate from IHIP, $\Delta r_s(n)$ forces CIHIP to converge to IHIP by removing the accumulated errors in the total height adjustment. In (5.11), $\Delta r_s(n)$ is a function of $d_s(n)$, representing that the more the user moves laterally, the more is the compensated height restored. The speed of restoration is determined by $a_s$, and it is chosen to be less than the difference threshold of slope perception reported in [16] to make the height restoration process imperceptible to the user. Combining the two terms, we obtain the following final update equation of $s(n)$:

$$s(n) = s(n-1) + \Delta s(n) + \Delta r_s(n). \tag{5.13}$$

Now, it is possible that $\mathbf{s}(n) \neq \mathbf{0}$, and so is $\mathbf{q}'(n) \neq \mathbf{q}(n)$. Hence, an additional collision detection test that uses CIHIP $\mathbf{q}'(n)$ is required to update the flag variable COLLISION. A total of six relative configurations are possible among HIP, IHIP, and CIHIP, as shown in Fig.5.6. In cases 1, 2, and 3 shown in the figure, HIP is under CIHIP, and thus a force should be rendered. These are equivalent to when $\mathbf{q}'(n) - \mathbf{p}(n)$ is in the same direction with the surface normal $\mathbf{n}(n)$. Therefore, the collision test returns TRUE if

$$(\mathbf{q}'(n) - \mathbf{p}(n)) \cdot \mathbf{n}(n) \geq 0. \tag{5.14}$$

Then, we compute the response force using (5.5) and set COLLISION to TRUE for the next frame.

Note that HIP can be outside of $M^{\mathbf{t}(n)}$ in case 3 of Fig.5.6. In this case, the regular collision detection algorithm between the trajectory of HIP and the mesh cannot detect a collision in the next frame $n+1$. Instead, we rely on the flag variable COLLISION set at $n$. If COLLISION is TRUE at $n+1$, we find $\mathbf{q}(n+1)$ to be the point that is constrained on the surface of $M^{\mathbf{t}(n+1)}$ and minimizes the distance from $\mathbf{p}(n+1)$ using the local graph search-based algorithm of Ho et al. [37].

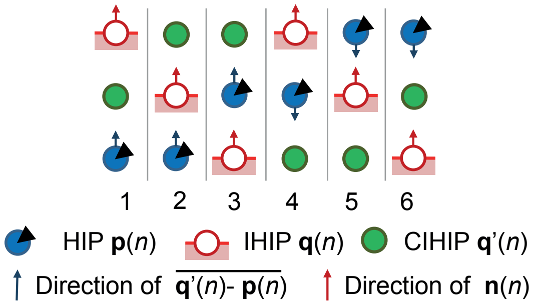


**Fig. 5.6** Possible configurations of HIP, IHIP, and CIHIP.

The procedure described in this subsection is repeated until COLLISION is changed to FALSE.

### 5.2.5 While Contact is Released

Suppose that the contact is released at $n_2$ and that this state continues until another contact is made at $n_3$ (Fig. 5.5d). At $n_2$, since a contact occurred in the previous frame, we can find $\mathbf{q}'(n_2)$ as described in Section 5.2.4. However, $\mathbf{q}'(n_2) - \mathbf{p}(n_2)$ is no longer in the same direction as $\mathbf{n}(n_2)$; therefore, the collision detection test returns FALSE. Then, we set COLLISION to FALSE and the rendering force to $\mathbf{0}$.

At this point, the point compensation vector $\mathbf{s}(n)$ stores the accumulated effect that is equivalent to translating the entire mesh $M$ by $\mathbf{s}(n)$. Thus, if the collision detection tests at $n > n_2$ use $M$ as the reference, the user may perceive the position difference between the meshes at $n_2$ and $n_3$. To prevent this problem, we use $M^{\mathbf{t}(n)}$ for collision detection, where $\mathbf{t}(n_2+1) = \mathbf{s}(n_2)$.

Analogous to the restoration of $\mathbf{s}(n)$, $\mathbf{t}(n)$ is decreased over time when the user moves in the free space, translating $M^{\mathbf{t}(n)}$ progressively back to $M$. This is to prevent the accumulation of $\mathbf{t}(n)$ over successive contacts and releases, since this accumulation may degrade the perception of the absolute position of $M$. Besides, this entire mesh translation allows us to use the same collision detection package throughout TCA-MESH.

The proposed restoration algorithm of $\mathbf{t}(n)$ is based on the observation that the more a user moves in space, the more s/he forgets the previous absolute location of the mesh. Specifically, while the non-contact state is maintained ($n_2 < n < n_3$), we compute a mesh restoration term $\Delta\mathbf{r}_t(n)$:

$$\Delta\mathbf{r}_t(n) = \begin{cases} -\mathbf{t}(n)\text{if } ||\mathbf{t}(n)|| \leq ||a_t d_t(n)|| \\ -a_t d_t(n)\dfrac{\mathbf{t}(n)}{||\mathbf{t}(n)||}\text{if } ||\mathbf{t}(n)|| > ||a_t d_t(n)|| \end{cases}, \tag{5.15}$$

where $d_t(n)$ denotes the displacement of HIP between the current and the previous frames:

$$d_t(n) = ||\mathbf{p}(n) - \mathbf{p}(n-1)||, \tag{5.16}$$

and $a_t$ is the mesh restoration ratio. The upper case in (5.15) is to make $\Delta\mathbf{r}_t(n) = \mathbf{0}$ when it is sufficiently small. Since this restoration process is performed while HIP is not contacting the surface, we can choose a relatively large value for $a_t$v as compared to $a_r$. The mesh compensation vector $\mathbf{t}(n)$ is updated as follows:

$$\mathbf{t}(n+1) = \mathbf{t}(n) + \Delta\mathbf{r}_t(n). \tag{5.17}$$

The mesh restoration is carried out only when COLLISION is FALSE, whereas the restoration of $\mathbf{s}(n)$ described in Section 5.2.4 is performed only during a contact state.

### 5.2.6 When another Contact Occurs

Finally, we describe the case when another contact occurs at $n = n_3$ (Fig. 5.5e). The computational procedure for response force computation is exactly the same as before, except that it may be the case that $M^{\mathbf{t}(n_3)} \neq M$ (Fig. 5.5f). Thus, $\mathbf{q}'(n)$ is calculated using both $\mathbf{s}(n)$ and $\mathbf{t}(n)$ with $\mathbf{t}(n) = \mathbf{t}(n_3)$, which preserves the translations that occurred during previous contacts.

### 5.2.7 Pseudocode

TCA-MESH explained above based on its flow can be summarized in a compact pseudocode. For implementation, readers can refer to Algorithm 1, where the collision detection–response structure is clearly visible.

**Algorithm 1** TCA-MESH

```
$\mathbf{p}(n)$, $\mathbf{p}(n-1)$, $\mathbf{t}(n)$, $k(n-1)$ and $s(n-1)$.
**if** COLLISION = FALSE **then**
    **if** $\mathbf{p}(n)-\mathbf{p}(n-1)$ overlaps with $M^{\mathbf{t}(n)}$ **then**
        Calculate $\mathbf{q}(n)$ on $M^{\mathbf{t}(n)}$.
        Get $k(n)$ and $\mathbf{n}(n)$ using $\mathbf{q}(n)$.
        $\mathbf{q}'(n) \leftarrow \mathbf{q}(n)$
        COLLISION $\leftarrow$ TRUE
    **else**
        Calculate $\Delta r_t(n)$ in (5.15).
        $\mathbf{t}(n+1) \leftarrow \mathbf{t}(n)+\Delta r_t(n)$
    **end if**
**else**
    Calculate $\mathbf{q}(n)$ on $M^{\mathbf{t}(n)}$.
    Get $k(n)$ and $\mathbf{n}(n)$ using $\mathbf{q}(n)$.
    Calculate $\Delta s(n)$ and $\Delta r_s(n)$ in (5.4) and (5.11), respectively.
    $s(n) \leftarrow \Delta s(n)+\Delta r_s(n)$
    $\mathbf{q}'(n) \leftarrow \mathbf{q}(n)+s(n)\mathbf{n}(n)$
    Update COLLISION by (5.14).
    **if** COLLISION = FALSE **then**
        $\mathbf{t}(n+1) \leftarrow \mathbf{t}(n)+s(n)\mathbf{n}(n)$
        $s(n) \leftarrow 0$
    **end if**
**end if**
**if** COLLISION = TRUE **then**
    $\mathbf{f}(n) \leftarrow k(n)(\mathbf{q}'(n)-\mathbf{p}(n))$
    $\mathbf{t}(n+1) \leftarrow \mathbf{t}(n)$
**else**
    $\mathbf{f}(n) \leftarrow \mathbf{0}$
**end if**
**return** $\mathbf{f}(n)$, $\mathbf{t}(n+1)$, $s(n)$ and $k(n)$.
```

## 5.3 Performance Evaluation: Trajectories

In this section, we present the probe trajectories, which correspond to the proximal stimuli for topography perception delivered to the user, when complex mesh surfaces with varying stiffness are explored for the experimental comparison between TCA-MESH and CBA. The three models shown in Fig. 5.3, captured and processed from real-world data, were used. All the models were scaled to fit into a 100 mm × 100 mm × 100 mm cube to match the workspace of the haptic interface (PHANToM Premium 1.5 High Force; Sensable Technologies, Inc.) used for data collection. The maximum stiffness of each model was 1.0 N/mm. For TCA-MESH, $a_s = 0.05$ and $a_t = 0.5$.

In both TCA-MESH and CBA, a probe trajectory is represented by that of HIP. In TCA-MESH, a force vector is directed from HIP to CIHIP, and its magnitude is proportional to the distance between them, while HIP and IHIP have the same role in CBA. We show the trajectories of these three major points in Fig. 5.7.

The first example is of the protein-on-mica model (Fig. 5.3a). This mesh with 131,072 triangles was generated from the height and stiffness maps captured using AFM [87]. With TCA-MESH, HIP well followed the surface of the protein-on-mica model, which coincided with the trajectory of IHIP, even in the lighter-colored, lower-stiffness halo region. In this region, the trajectory of CIHIP deviated from the surface to compensate for the stiffness changes. In the case of CBA, however, HIP dropped at the beginning of the halo region (marked by point A in Fig. 5.7a) and then abruptly elevated at the end of the halo region (point B in Fig. 5.7a), reconfirming the problem reported in [96, 18].

The second example is of the Stanford bunny model (Fig. 5.3a) with 69,451 triangles, which was reconstructed from 3D scan data. In the original mesh, each vertex stored the confidence of the range data. We mapped these confidence values to stiffness values, with a maximum stiffness of 1.0 N/mm. This example demonstrates a case in which two independent types of information are separately embedded into two haptic attributes, with a higher priority on shape rendering. In Fig.5.7b, the region enclosed by a red dotted line is concave, and its stiffness value is lower than that of the neighborhood. The trajectory of

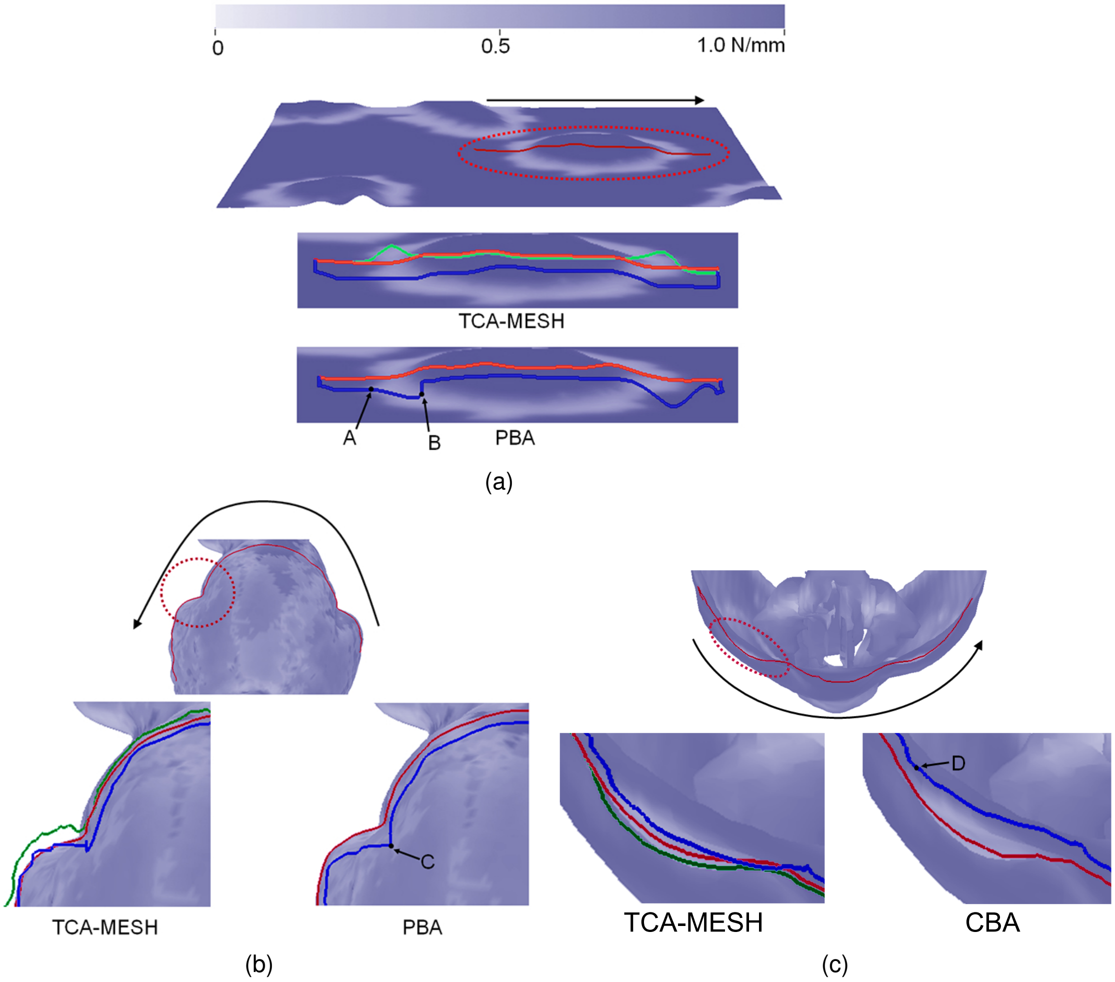


**Fig. 5.7** Trajectories of the major points with TCA-MESH and CBA when the user strokes three models: (a) protein-on-mica (red dotted circle: elevated area), (b) Stanford bunny (red dotted circle: concave area), and (c) Human head (red dotted ellipse: eye). The trajectories of HIP, IHIP, and CIHIP are represented by blue, red, and green solid lines, respectively. An arrow-headed line indicates the stroke direction. The models and trajectories consisted of 3D data, and hence, they are rendered here using a perspective view. All models share the same color map for stiffness shown on the top of (a). Each figure includes trajectories magnified from the region enclosed by a red dotted line.

HIP with TCA-MESH well followed this complex geometry, showing a small jump when the probe hit point C. However, CBA let HIP be dipped into the bunny and failed to deliver the stronger sensation of contact at point C.

The last example is of the human head model (Fig. 5.3c) with 57,608 triangles. This model shows an isosurface of the human skin reconstructed using the VTK toolkit from the CT volumetric data included in the toolkit as a medical example [97]. This is another example in which shape is of more importance. The stiffness value (0.4–1.0 N/mm) of each vertex is inversely proportional to the distance between the skin and the underlying bone. The model includes an eye (red dotted ellipse, Fig. 5.7c) which protrudes from the surrounding skin and has lower stiffness. In the case of TCA-MESH, the HIP followed the protruded eye skin adequately, but in the case of CBA, the HIP showed increasing penetration at point D.

## 5.4 Performance Evaluation: User Study

This section reports two perceptual experiments for shape and stiffness discrimination, respectively, carried out to assess the performance improvement of TCA-MESH over CBA.

### 5.4.1 Methods

#### Participants

Sixteen participants (14 males and 2 females; 19-27 years old with a mean of 21.3 years) participated in both experiments. All participants were right-handed and reported no known sensorimotor impairment. None had prior experience of using a haptic interface to explore virtual objects.

#### Apparatus

An impedance-type haptic interface (PHANToM Premium 1.5 High Force; Sensable Technologies, Inc.) with a stylus tool and a 22-inch LCD monitor were used for haptic and visual rendering, respectively. A barrier was placed between the haptic interface and the

**Table 5.1** Shape-stiffness models used in the perceptual evaluation.

| | Shape | | | | |
|---|---|---|---|---|---|
| Stiffness | H0: plane | H1: bump | H2: bump | H3: hole | H4: hole |
| K0: plane | M0 | M5 | M10 | M15 | M20 |
| K1: softer center | M1 | M6 | M11 | M16 | M21 |
| K2: softer center | M2 | M7 | M12 | M17 | M22 |
| K3: harder center | M3 | M8 | M13 | M18 | M23 |
| K4: harder center | M4 | M9 | M14 | M19 | M24 |

monitor to prevent the participants from obtaining additional visual cues by watching the movements of the PHANToM.

### Experimental Conditions

To compose virtual 3D mesh surfaces to be used in the experiments, we designed five implicit shape models and five implicit stiffness models. As shown in Fig. 5.8, the shape models were a flat surface (H0), two bumps (H1 and H2), and two holes (H3 and H4). The stiffness models were a constant stiffness model (K0) and four non-uniform stiffness distributions (K1–K4). By combining the five shape and the five stiffness models, we obtained 25 models (M0–M24), as listed in Table 5.1.

These 25 shape-stiffness models were then converted into 25 individual 3D meshes. The geometry and the stiffness of each model were densely and evenly sampled, and mapped onto the vertices of a corresponding mesh. A final virtual surface was constructed by embedding the 25 meshes onto a flat surface in a 5×5 array, as shown in Fig. 5.9a. This virtual surface consisted of a total of 39,200 triangles, representing a complex dataset in which a number of objects were included with spatially-varying stiffness. Note that the mesh models using H2 and H4 included sharp edges in their geometry to see whether TCA-MESH can handle sharp normal changes. The location of each shape-stiffness model on the surface was randomized in each experimental trial.

Out of the 25 models, 5 models (M0, M5, M10, M15, and M20) had a constant stiffness distribution (K0). Their shapes were generally perceived without distortion. The other

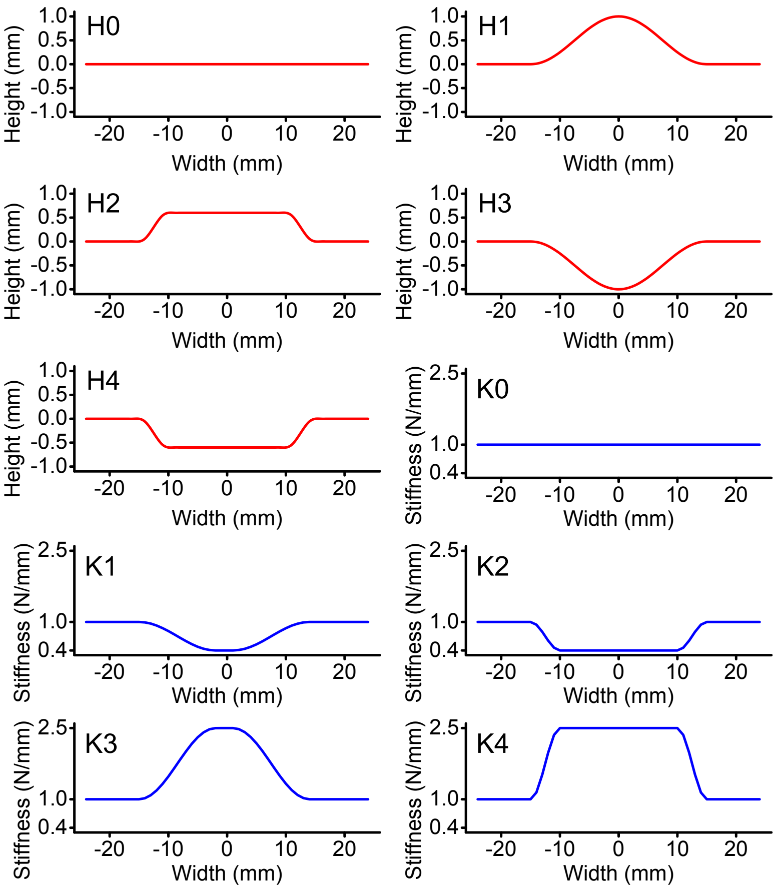


**Fig. 5.8** Five shape (H0–H4) and five stiffness models (K0–K4) used in the perceptual performance evaluation.

20 models had non-uniform stiffness distributions, which could cause distortion in shape perception when rendered with CBA. In particular, we expected that 12 of them could lead to an incorrect classification of the shape. In eight models (M6, M7, M11, M12, M18, M19, M23, and M24), a bump could be perceived as a hole, or vice versa. The other four flat models (M1–M4) could be regarded as a bump or a hole depending on the paired stiffness model (K1–K4). In terms of stiffness perception, we expected that all models would lead to correct classification irrespective of the rendering method.

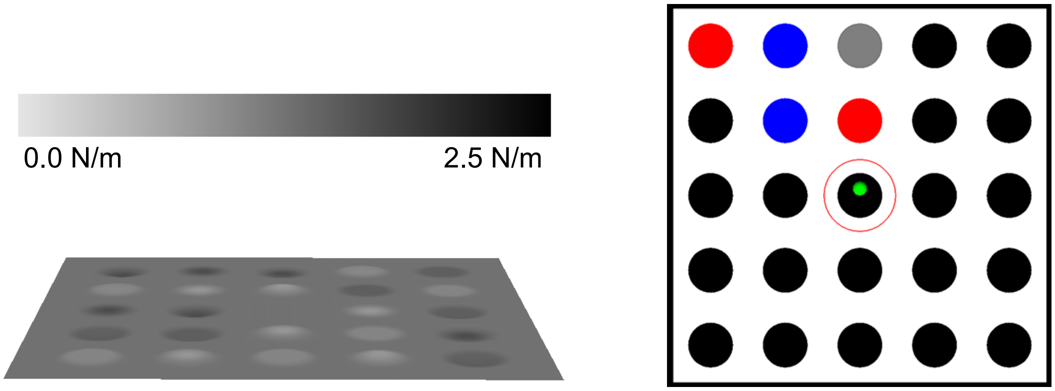
### Procedures

The participant's task in each trial was as follows. The participant was provided with the visual scene shown in Fig. 5.9b. The locations of the 25 models were represented by solid circles that were initially black in color. The 2D location of the HIP was drawn as a small green circle. When the HIP was sufficiently close to a model, a larger red circle was drawn around the corresponding circle. To preclude any visual topography cues, the scene was rendered using orthogonal projection without shading, and the HIP was always visible even if it was occluded by the mesh.

The participant was instructed to explore the virtual surface continuously, from one model to another, using *any EPs*, without a time limit. For shape discrimination, the participant was asked to perceive the shape of each model and classify it as a bump, hole, or plane. For stiffness discrimination, the participant was asked to compare the stiffness at the center of each model with that of the surroundings, and then label the model as 'harder' (center stiffer than the surroundings), 'softer,' or 'the same.' In each case, the circle was then colored as red, blue, or gray, respectively. The participant used a keyboard to label the models. The trial was completed after all the models had been labeled.

Each experiment was composed of one training session and two main sessions. The purpose of the training session was to familiarize the participants with their task. Therefore, all shape-stiffness models in this session were of constant stiffness (K0) for shape discrimination or flat (H0) for stiffness discrimination, not the full set described in Section 5.4.1. The training session had two trials. In the first trial, the virtual surface included only 3×3 models for easy practice. In the second trial, the surface included 5×5 models. The virtual surface was rendered using CBA.

Each of the two main sessions consisted of five trials. For rendering, TCA-MESH was used in one main session, whereas CBA was used in the other. Their order was balanced across the participants to prevent any order effects. The height restoration constant, $a_s$, used for TCA-MESH was 0.05, which was lower than the minimum (0.09) of the various absolute thresholds of the slope perception reported in [16]. The mesh restoration constant, $a_t$, was 0.5. In each trial, the full 25 shape-stiffness models in Table 5.1 were used to make



(a) (b)

**Fig. 5.9** (a) Mesh surface embedding the 25 shape-surface models. (b) Visual scene provided to the participant.

a virtual surface.

To prevent fatigue, the participants were required to take at least a 1-min break between trials and at least a 3-min break between sessions.

After each trial, the program recorded the number of correct classifications and the task completion time. The participants were also asked to complete a questionnaire that included four questions. Three questions were descriptive and were answered after each main session: Q1: When you perceived the shape/stiffness, what sensation was the most salient to you?; Q2: To discriminate the shape/stiffness, what sensation did you focus on?; and Q3: Did you feel anything unnatural? If yes, describe when you felt so. The fourth question was answered after each shape/stiffness discrimination experiment: Q4: Between the two haptic rendering algorithms used in the two main sessions, which one do you prefer for shape/stiffness discrimination?

The shape discrimination experiment took approximately 1.5 h, and the stiffness discrimination experiment took approximately 1 h. Both experiments were completed in one day, and the shape discrimination experiment was conducted first.

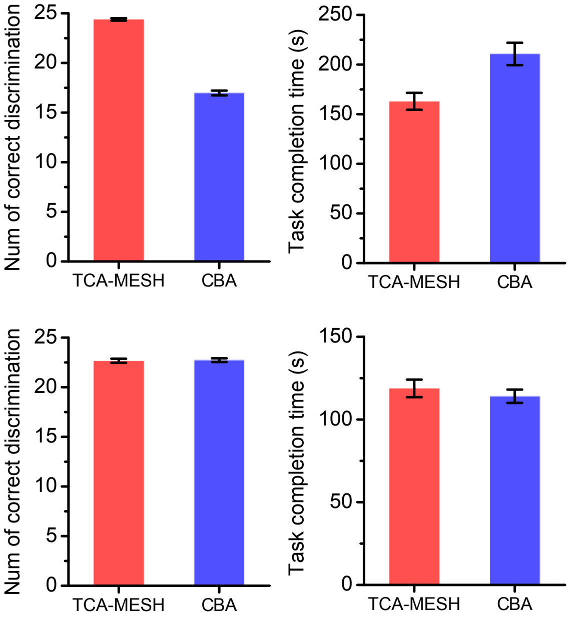


**Fig. 5.10** Quantitative experimental results. Error bars represent standard errors.

### 5.4.2 Results

The main results of the two experiments are shown in Fig. 5.10. For shape discrimination, the average number of correct classifications was 24.4 ± 1.1 (standard deviation) in the case of TCA-MESH and 17.0 ± 2.2 in the case of CBA (left graph, Fig. 5.10a). For statistical testing, we used the Friedman test since the numbers of correct classifications were not normally distributed. The difference was statistically significant ($\chi^2(1) = 117.07$, $p < 0.0001$). The average task completion times were 163 ± 76 s and 211 ± 101 s in the case of TCA-MESH and CBA, respectively (right graph, Fig. 5.10a). This difference, tested using ANOVA, was also statistically significant ($F(1{,}15) = 7.98$, $p = 0.0181$).

As for stiffness discrimination, the average numbers of correct classifications were 22.7 ± 1.8 using TCA-MESH and 22.7 ± 1.7 using CBA (left graph in Fig. 5.10b). This difference was not statistically significant (Friedman test, $\chi^2(1) = 0.1378$, $p = 0.7105$). The average task completion times were 118.8 ± 47.5 s using TCA-MESH and 114.1 ± 36.1 s using

CBA (right graph in Fig. 5.10b). This difference was not statistically significant (ANOVA, $F(1,15) = 0.26$, $p = 0.6177$).

In the case of shape discrimination, the participants' responses to Q1 and Q2 of the questionnaire could be divided into two groups. One group (G1) picked the position changes of the PHANToM stylus to both questions. The other group (G2) provided diverse answers, but the members of the group commonly mentioned some changes in the response force, e.g., in the force direction, tangential force, or normal force. In the case of TCA-MESH, the numbers of participants in G1 were 9 for Q1 and 8 for Q2, and in G2, they were 7 for Q1 and 8 for Q2. In the case of CBA, the numbers in G1 were 10 for Q1 and 9 for Q2, and in G2, they were 6 for Q1 and 7 for Q2. In the response to Q3, only 4 participants reported that they felt unnatural artifacts with the virtual surface rendered using TCA-MESH, but 10 participants reported such artifacts in the case of CBA. In the response to Q4, 8 participants chose TCA-MESH as the method that they would prefer for shape discrimination, 4 participants selected CBA, and the other 4 participants had no preference.

In the case of stiffness discrimination, the participants' answers to Q1 and Q2 for stiffness discrimination also showed two patterns. One group of the participants (G3) reported that in the presence of a stiffness change, they felt a position or velocity difference when they applied the same force to the surface. The other group (G4) relied on the magnitude difference of the response force. The numbers of the participants in G3 and G4 were 11 and 5 for both Q1 and Q2 in the case of TCA-MESH. They were 10 and 6 for both Q1 and Q2 with CBA. In the response to Q3, 7 participants reported an unnatural sensation in the case of CBA, but the number of such participants was 1 in the case of TCA-MESH. In the response to Q4, 8 participants preferred TCA-MESH for stiffness discrimination, 4 participants preferred CBA, and the other 4 participants had no preference.

### 5.4.3 Discussion

For shape perception, TCA-MESH exhibited a considerably improved performance as compared to CBA with statistical significance in both the number of correct classifications (24.4 vs. 17.0) and the task completion time (163 s vs. 211 s). In particular, the number of cor-

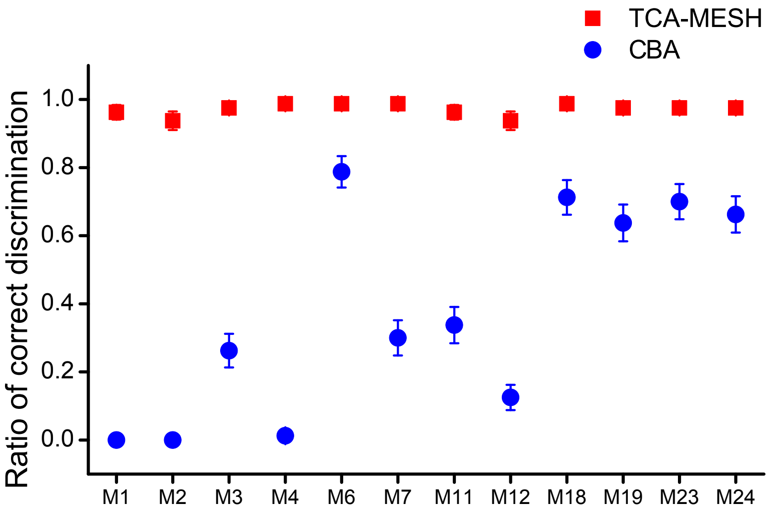


**Fig. 5.11** Ratios of correct shape classifications for the 12 models that have a high probability of incorrect discrimination when CBA is used. Error bars represent standard errors.

rect classifications of TCA-MESH was very close to the perfect score (25), indicating the effectiveness of TCA-MESH.

In general, the two sensory cues, i.e., changes in probe position and changes in response force direction, can contribute to shape discrimination [15, 26]. This general fact is consistent with our experimental results. While TCA-MESH renders both sensory cues to be consistent, CBA may cause a conflict between the two cues. The response force direction of CBA always delivers a helpful cue, but its probe position may interfere with accurate shape discrimination [15]. This can also be confirmed by the number of correct classifications for the 12 models that were expected to lead to incorrect discrimination in the case of CBA, as shown in Fig. 5.11. While the correct classification ratios of TCA-MESH were very close to 1 for all of these 12 models, those of CBA were not. In particular, M1-M4 resulted in very low correct classification ratios when CBA was used. These models had a flat surface (H0) combined with one of the four non-uniform stiffness models (K1–K4). In this case, the response force direction did not change; thus, the participants had to rely only on the probe position changes for shape discrimination, causing such low correct classification ratios.

For the bumps and holes, however, some participants could discriminate the shape; see

M6, M18, M19, M23, and M24 in Fig. 5.11. This could be attributed to the fact that participants can also use the force directional cue for shape discrimination when they are unable to make reliable discrimination based on the position cue. Research demonstrated that the force cue could overcome the position cue for shape discrimination [82]. This was also supported by our questionnaire results for Q1 and Q2: the participants' preference for using one of the two sensory cues as the main criterion was close to even.

For stiffness perception, both TCA-MESH and CBA showed a high number of correct classifications (both 22.7) with comparable task completion times (118.8 and 114.1 s) without any statistically significant difference. This indicates that TCA-MESH has high stiffness rendering accuracy comparable to that of CBA. For both methods, most misclassifications occurred with the constant-stiffness models (M0, M5, M10, M15, and M20). This result of stiffness perception is despite the fact that TCA-MESH masks the relative changes in stiffness during stroking for shape compensation. This in turn implies that the participants mostly tapped on the virtual surface to gauge stiffness differences, which is consistent with the theory of EPs [60]. The experimenter observed that stroking was used predominantly in the shape discrimination experiments and that tapping was used predominantly in the stiffness discrimination experiment, although the participants were allowed to use any EPs. This could be another evidence of the adequacy of TCA-MESH for multi-attribute haptization.

Lastly, we note that the task completion time of TCA-MESH (163 s on average) demonstrates that TCA-MESH could be used stably for a long time (>2 min). This supports that the proposed height restoration strategy using the height restoration term $\Delta r_s(n)$ and the mesh restoration term $\Delta \mathbf{r}_t(n)$ is appropriate.

Chapter 6

# Conclusion

This dissertation presents two core examples for modeling and transferring such inhomogeneous haptic data.

Our first example focuses on capturing and rendering the behaviors of real objects of inhomogeneous deformation dynamics. We adapted a framework of the "data-driven haptics," where the response forces are modeled based on the recorded haptic data, and they are reproduced using an interpolation schemes. We first have presented a hatic shape modeling system that provides the shape modeling ability to a regular force-feedback haptic interface device, even for soft and deformable object. The user can sample points on an object surface using the haptic device based on their own discretion. The deformation errors of the object are compensated for by our true contact point estimation algorithm. The collected point set is processed by the standard alpha shape algorithm for mesh reconstruction. Experimental results indicated that our modeling system can lead to comparable performance to a standard 3D optical scanner for a range of real objects, as long as they do not contain fine surface details. Our system is free of the optical properties of real objects and only requires a force sensor in addition to a common haptic interface. Then, with the shape model captured by shape modeling system, we can model the reaction force behaviors of inhomogeneous soft-deformable object. The core point is the simulation of proxy point (actual contact point) movement based on sliding yield surface models, which possess necessary information for separating sliding and sticking states. In an off-line process, sliding yield surface models and internal radial-basis models are built through an automatically palpation of a target real object. During rendering, the movement of a proxy point is estimated using the sliding yield surface models, which becomes an input parameter of the radial basis interpolation models. Allowing to user the data-acquisition in the unconstrained exploration is unique aspect of our modeling framework. In the performance evaluation, our framework shows less than 0.5N force error ratio in most cases, both in normal lateral direction. Our modeling system does not limit the exploratory procedure in not only data acquisition but also rendering time.

The second example of this dissertation presents a computational algorithm for perceptually-correct haptization of inhomogeneous scientific data captured from the real environment. We have presented TCA-MESH, a haptization algorithm of object shape represented by a 3D mesh with a spatially varying stiffness distribution, motivated by the need for scientific data perceptualization. TCA-MESH is based on the human exploratory behavior of force constancy, which allows us to estimate the appropriate amount of height compensation on the basis of the user-applied force and stiffness difference. TCA-MESH also provides an improved estimator of the user-applied force to handle complex mesh models that may include a number of sharp normal direction changes around the edges. We demonstrated the enhanced performance TCA-MESH by using the relevant proximal stimuli recorded during the exploration of complex mesh objects. We also conducted two perceptual experiments, which demonstrated that TCA-MESH enables the user to accurately perceive both the shape of a virtual surface and its varying stiffness distribution. TCA-MESH exhibited a higher accuracy and required a considerably shorter task completion time than the conventional constraint-based rendering. In addition, it did not impart noticeable artifacts and worked stably over prolonged use. A mesh is a general geometry model for describing a complex 3D surface and is widely used for computer graphics, data visualization, and medical imaging. As such, we expect that TCA-MESH can be applied to various application areas. Moreover, it can be easily extended to other surface representations in part because of the use of an ordinary collision detection algorithm that can be the most troublesome impediment to haptic rendering. One current limitation of TCA-MESH is that the mesh compensation vector can largely deviate from the orig- inal mesh position in extreme cases. In general use, the mesh restoration term can prevent that problem, as demonstrated in the user study. This limitation might be resolved if the stiffness model is modulated instead of the shape model. This is one of the future research topics in our mind, along with an extension to volumetric models with internal stiffness definitions.

# Bibliography

[1] S. Allin, Y. Matsuoka, and R. Klatzky. Measuring just noticeable differences for haptic force feedback: implications for rehabilitation. In *Proceedings of Haptics Symposium*, pages 299–302, 2002.

[2] R. Avila and L. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the ACM Conference on Visualization*, pages 197–204, 1996.

[3] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. Srinivasan. Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64, 2004.

[4] F. Bernardini and C. L. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proceedings of the Canadian Conference on Computational Geometry*, pages 193–198, 1997.

[5] F. Bernardini, C. L. Bajaj, J. Chen, and D. R. Schikore. Automatic reconstruction of 3D CAD models from digital scans. *International Journal of Computational Geometry and Applications*, 9(4):327–369, 1999.

[6] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002.

[7] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[8] R. Bowen Loftin. Multisensory perception: beyond the visual in visualization. *Computing in Science Engineering*, 5(4):56–58, 2003.

[9] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490–503, 1998.

[10] F. P. Brooks, Jr., M. Ouh-Young, J. J. Batter, and P. Jerome Kilpatrick. Project GROPE - Haptic displays for scientific visualization. In *Proceedings of the ACM SIGGRAPH Conference*, pages 177–185, New York, NY, USA, 1990.

[11] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory,*, 52(12):5406–5425, 2006.

[12] A. Cantoni. Optimal curve fitting with piecewise linear functions. *IEEE Transactions on Computers*, C-20(1):59 – 67, 1971.

[13] F. Cazals and M. Pouget. Algorithm 889: Jet fitting 3 : A generic C++ package for estimating the differential properties on sampled surfaces via polynomial fitting. *ACM Transactions on Mathematical Software*, 35:24:1–24:20, 2008.

[14] J. Cheon and S. Choi. Perceptualizing a “haptic edge” with varying stiffness based on force constancy. *Lecture Notes on Computer Science (ICAT 2006)*, 4282:392–405, 2006.

[15] J. Cheon and S. Choi. Haptizing a surface height change with varying stiffness based on force constancy: Effect of surface normal rendering. In *Proceedings of the World Haptics Conference*, pages 84–89, 2007.

[16] J. Cheon, I. Hwang, K. Han, and S. Choi. Haptizing a surface height change with varying stiffness based on force constancy: Extended algorithm. In *Proceedings*

*of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 193–200, 2008.

[17] V. S. Chib, J. L. Patton, K. M. Lynch, and F. A. Mussa-Ivaldi. Haptic identification of surfaces as fields of force. *Journal of Neurophysiology*, 95(2):1068–1077, 2006.

[18] S. Choi, L. Walker, H. Z. Tan, S. Crittenden, and R. Reifenberger. Force constancy and its effect on haptic perception of virtual surfaces. *ACM Transactions on Applied Perception*, 2(2), 2005.

[19] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[20] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 140–145, 1995.

[21] M. B. Colton and J. M. Hollerbach. Haptic models of an automotive turn-signal switch: Identification and playback results. In *Proceedings of World Haptics Conference (WHC)*, pages 243–248. IEEE, 2007.

[22] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):62–73, Jan 1999.

[23] A. Curnier. A theory of friction. *International Journal of Solids and Structures*, 20(7):637 – 647, 1984.

[24] P. R. Dahl. A solid friction model. Technical report, Aerospace Corporation El Segundo CA., 1968.

[25] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.

[26] K. Drewing and M. O. Ernst. Integration of force and position cues for shape perception through active touch. *Brain Research*, 1078(1):92–100, 2006.

[27] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47, 2006.

[28] N. Durlach, L. Delhorne, A. Wong, W. Ko, W. Rabinowitz, and J. Hollerbach. Manual discrimination and identification of length by the finger-span method. *Attention, Perception, & Psychophysics*, 46:29–38, 1989.

[29] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. In *Proceedings of the Workshop on Volume Visualization*, pages 75–82, 1992.

[30] T. Edmunds and D. K. Pai. Perceptual rendering of learning haptic skills. In *Proceedings of Haptics Symposium*, pages 225–230. IEEE, 2008.

[31] A. Fabri and S. Pion. CGAL: The computational geometry algorithms library. In *Proceedings of the ACM International Conference on Advances in Geographic Information Systems*, pages 538–539, 2009.

[32] P. Fong. Sensing, acquisition, and interactive playback of data-based models for elastic deformable objects. *The International Journal of Robotics Research*, 28(5):630–655, 2009.

[33] A. Formaglio, G. Baud-Bovy, and D. Prattichizzo. Conveying virtual tactile feedback via augmented kinesthetic stimulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3995–4000, 2007.

[34] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: a hierarchical structure for rapid interference detection. In *Proceedings of the ACM SIGGRAPH Conference*, pages 171–180, 1996.

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

[36] K. Hashiguchi. *Elastoplasticity theory*, volume 42. Springer, 2009.

[37] C. Ho, C. Basdogan, and M. A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoperators and Virtual Environment*, 8(5):477–491, 1999.

[38] R. Hoever, G. Kosa, G. Székely, and M. Harders. Data-driven haptic rendering – From viscous fluids to visco-elastic solids. *IEEE Transactions on Haptics*, 2:15–27, 2009.

[39] R. Hover, M. Di Luca, G. Szekely, and M. Harders. Computationally efficient techniques for data-driven haptic rendering. In *Proceedings of World Haptics*, pages 39–44, March 2009.

[40] R. Hover, G. Kosa, G. Szekely, and M. Harders. Data-driven haptic rendering - from viscous fluids to visco-elastic solids. *IEEE Transactions on Haptics*, 2(1):15–27, Jan 2009.

[41] K. H. Huebner, D. L. Dewhirst, D. E. Smith, and T. G. Byrom. *The Finite Element Method for Engineers*. John Wiley & Sons, 2008.

[42] K. Hunt and F. Crossley. Coefficient of restitution initerpreted as damping in vibroimpact. *ASME Journal of Applied Mechanics*, 42:440–445, 1975.

[43] M. Ikits, J. Brederson, C. Hansen, and C. Johnson. A constraint-based technique for haptic volume exploration. In *Proceedings of the IEEE Visualization Conference*, pages 263–269, 2003.

[44] H. Iwata and H. Noma. Volume haptization. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 16–23, 1993.

[45] D. L. James and D. K. Pai. Artdefo: Accurate real time deformable objects. In *Proceedings of the ACM SIGGRAPH*, pages 65–72, 1999.

[46] F. Janabi-Sharifi, V. Hayward, and C. S. J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, 8(6):1003–1009, 2000.

[47] S. Jeon and S. Choi. Haptic augmented reality: Taxonomy and an example of stiffness modulation. *Presence: Teleoperators and Virtual Environments*, 18(5):387–408, 2009.

[48] S. Jeon and S. Choi. Real stiffness augmentation for haptic augmented reality. *Presence: Teleoperators and Virtual Environments*, 20(4), 2011.

[49] S. Jeon, J.-C. Metzger, S. Choi, and M. Harders. Extensions to haptic augmented reality: Modulating friction and weight. In *Proceedings of the IEEE World Haptics Conference (WHC)*, pages 227–232, 2011.

[50] M. Judd, T. Brown, and M. Colton. Toward automated haptic modeling using commercial haptic interfaces: surface normal estimation and static model identification. In *Proceedings of World Haptics 2009*, pages 434–439, March 2009.

[51] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.

[52] N. Kikuchi and J. T. Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods*, volume 8. Siam, 1988.

[53] M. Kilchenman O'Malley and M. Goldfarb. Comparison of human haptic size identification and discrimination performance in real and simulated environments. In *Proceedings of the International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 10–17, 2002.

[54] L. Kobbelt. Sqrt(3)-subdivision. In *Proceedings of the ACM SIGGRAPH*, pages 103–112, 2000.

[55] P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Transaction on Graphics*, 25(3):872–880, July 2006.

[56] J. Lang, D. K. Pai, and R. J. Woodham. Acquisition of elastic models for interactive simulation. *The International Journal of Robotics Research*, 21(8):713–733, 2002.

[57] J. Lang and A. S. Measurement-based modeling of contact forces and textures for haptic rendering. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):380–391, 2011.

[58] D. A. Lawrence, C. D. Lee, L. Y. Pao, and R. Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In *Proceedings of the conference on Visualization*, pages 131–137. IEEE Computer Society Press, 2000.

[59] D. A. Lawrence, L. Y. Pao, C. D.Lee, and R. Y. Novoselov. Synergistic visual/haptic rendering modes for scientific visualization. *IEEE Computer Graphics and Applications*, 24(6):22–30, 2004.

[60] S. J. Lederman and R. L. Klatzky. Hand movement: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.

[61] D. J. Lee. Extension of Colgate's passivity condition to variable-rate haptics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1761–1766, 2009.

[62] A. Leeper, S. Chan, and K. Salisbury. Constraint-based 3-DOF haptic rendering of arbitrary point cloud data. In *RGB-D 2011 Workshop in Conjuction with RSS*, Los Angeles, CA, June 2011.

[63] M. C. Lin and W. V. Baxter. Modeling and creative processes. In M. C. Lin and M. A. Otaduy, editors, *Haptic Rendering Foundations, Algorithms, and Applications*, chapter 28, pages 531–548. A K Peters/CRC Press, 2008.

[64] K. Lundin, M. Cooper, A. Persson, D. Evestedt, and A. Ynnerman. Enabling design and interactive selection of haptic modes. *Virtual Reality*, 11:1–13, 2007.

[65] K. Lundin, B. Gudmundsson, and A. Ynnerman. General proxy-based haptics for volume visualization. In *Proceedings of the World Haptics conference*, pages 557–560, 2005.

[66] K. Lundin, A. Ynnerman, and B. Gudmundsson. Proxy-based haptic feedback from volumetric density data. In *Proceedings of the Eurohaptics*, pages 104–109, 2002.

[67] R. Maciejewski, S. Choi, D. Ebert, and H. Tan. Multi-modal perceptualization of volumetric data and its application to molecular docking. In *Proceedings of the World Haptics Conference*, pages 511–514, 2005.

[68] K. E. MacLean. The ”Haptic Camera”: A technique for characterzing and playing back haptic properties of real environments. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 459–467, 1996.

[69] R. Michalowski and Z. Mroz. Associated and non-associated sliding rules in contact friction problems. *ARCH. MECH. STOSOW*, (30):259–276, 1978.

[70] H. Morgenbesser and M. Srinivasan. Force shading for haptic shape perception. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume 58, pages 407–412. American Society of Mechanical Engineers, 1996.

[71] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[72] A. Okamura, I. Webster, R.J., J. Nolin, K. Johnson, and H. Jafry. The haptic scissors: cutting in virtual environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 828–833 vol.1, Sept 2003.

[73] A. M. Okamura, K. J. Kuchenbecker, and M. Mahvash. *Haptic Rendering Foundations, Algorithms, and Applications*, chapter Measurement-Based Modeling for Haptic Rendering, pages 443–548. A K Peters/CRC Press, 2008.

[74] M. A. Otaduy and M. C. Lin. *Haptic Rendering: Foundations, Algorithms and Applicationsis*, chapter Introduction to Haptic Rendering Algorithm, pages 159–180. A K Peters/CRC Press, 2008.

[75] S. Ozaki, K. Hikida, and K. Hashiguchi. Elastoplastic formulation for friction with orthotropic anisotropy and rotational hardening. *International Journal of Solids and Structures*, 49(3-4):648 – 657, 2012.

[76] D. K. Pai and P. Rizun. The WHaT: A wireless haptic texture sensor. In *Proceedings of the International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 3–9, 2003.

[77] D. K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau. Scanning physical interaction behavior of 3D objects. In *Proceedings of the ACM SIGGRAPH*, pages 87–96, 2001.

[78] K. L. Palmerius. Fast and high precision volume haptics. In *Proceedings of the World Haptics Conference*, pages 501–506, 2007.

[79] K. L. Palmerius. Adding tangential forces in lateral exploration of stiffness maps. In *Lecture Notes in Computer Science (HAID 2011)*, volume 6851, pages 1–10, 2011.

[80] K. L. Palmerius and C. Forsell. The impact of feedback design in haptic volume visualization. In *Proceedings of the World Haptics Conference*, pages 154 –159, 2009.

[81] S. Paneels and J. Roberts. Review of designs for haptic data visualization. *IEEE Transactions on Haptics*, 3(2):119–137, 2010.

[82] G. Robles-De-La-Torre and V. Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412:445–448, 2001.

[83] J. Romano, T. Yoshioka, and K. Kuchenbecker. Automatic filter design for synthesis of haptic textures from recorded acceleration data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1815–1821, 2010.

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

[85] F. Rydén, S. Nia Kosari, and H. Chizeck. Proxy method for fast haptic rendering from time varying point clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2614–2619. IEEE, 2011.

[86] J. Ryu and S. Choi. posVibEditor: Graphical authoring tool of vibrotactile patterns. In *Proceedings of the IEEE International Workshop on Haptic, Audio and Visual Environments and Games (HAVE)*, pages 120–125, 2008.

[87] D. Sarid. *Scanning Force Microscopy*. Oxford Unversity Press, New York, 1991.

[88] J. L. Schoner, J. Lang, and H.-P. Seidel. Measurement-based interactive simulation of viscoelastic solids. In *Proceedings of Computer Graphics Forum*, volume 23, pages 547–556. Wiley Online Library, 2004.

[89] A. Sianov and M. Harders. Data-driven haptics: Addressing inhomogeneities and computational formulation. In *World Haptics Conference*, pages 301–306. IEEE, 2013.

[90] Stanford University Computer Graphics Laboratory. Stanford Bunny. http://graphics.stanford.edu/data/3Dscanrep/.

[91] C. Swindells, E. Maksakov, K. E. Maclean, and V. Chung. The role of prototyping tools for haptic behavior design. In *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 161–168, 2006.

[92] H. Z. Tan, M. A. Srinivasan, B. Eberman, and B. Cheng. Human factors for the design of force-reflecting haptic interfaces. In *Proceedings of the 3rd International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 353–359. ASME, 1994.

[93] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the ACM SIGGRAPH*, pages 351–358, 1995.

[94] R. M. Taylor, W. Robinett, V. L. Chi, F. P. Brooks, Jr., W. V. Wright, R. S. Williams, and E. J. Snyder. The nanomanipulator: a virtual-reality interface for a scanning tunneling microscope. In *Proceedings of the ACM SIGGRAPH Conference*, pages 127–134, 1993.

[95] M. Vicentini and D. Botturi. Human factors in haptic contact of pliable surfaces. *Presence: Teleoperators and Virtual Environments*, 18(6):478–494, 2009.

[96] L. Walker and H. Tan. A perceptual study on haptic rendering of surface topography when both surface height and stiffness vary. In *Proceedings of Haptics Symposium*, pages 138–145, March 2004.

[97] B. L. Will Schroeder, Ken Martin. *The Visualization Toolkit, 4th Edition*. Kitware, Inc., 2006.

[98] P. Wriggers and T. A. Laursen. *Computational contact mechanics*, volume 30167. Springer, 2006.

[99] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, July 2009.

[100] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura. Tissue property estimation and graphical display for teleoperated robot-assisted surgery. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3117–3123, 2009.

[101] H. Yano, M. Nudejima, and H. Iwata. Development of haptic rendering methods of rigidity distribution for tool-handling type haptic interface. In *Proceedings of the World Haptics Conference*, pages 569–570, 2005.

[102] C. Zilles and J. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 146–151, 1995.

# 요 약 문

## 비균일 햅틱 정보의 모델링 및 전달

많은 햅틱 응용 프로그램에서 비균일한 햅틱 정보를 다루는 것은 중요한 요소이다. 특히 의생물 분야의 정보는 본질적으로 비균일하며 의료 햅틱 응용에서는 비균일한 정보 사이의 상호작용으로 발생하는 정보를 모델링하고 재생할 수 있는 복잡하고 효좌적인 햅틱 알고리즘을 필요로 한다. 이 박사학위 논문에서는 비균일한 햅틱 정보를 다루고 전달하는 두 가지 주요한 응용 분야를 다루기로 한다.

우리는 먼저 비균일한 동적 변형을 보여주는 실제 물체의 거동을 모델링하고 렌더링하는 것에 초점을 두었다. 이를 위해 우리는 실제 물체로 부터 측정하고 저장한 햅틱 정보를 바탕으로 하고 이를 보간하여 제시하는 “data-driven haptics” 방법론을 이용하였다. 이 방법론은 다양한 범위의 물리적인 현상을 하나의 프레임 워크로 다룰 수 있다. 우리의 프레임 워크는 특히 수평적인 방향의 비균일적인 성질, 즉 마찰과 제한 되지 않은 거동 즉 미끄지거나 문지르는 거동을 정확하게 측정하고 렌더링 할 수 있었으며, 이는 햅틱 모델링 및 렌더링 방법론에 있어 첫번째 시도이다. 중요한 요소는 물체와 프로브 사이의 sliding/stick state를 표현하는 slding yield surface를 이용한 proxy 거동의 시뮬레이션이다. 전처리 과정 동안 sliding yield surface는 자동적으로 물체와 상호작용하며 작성된다. 내부적인 radial-basis function도 전처리 과정동안 계산된다. 렌더링 시에 proxy의 위치는 sliding yield surface 모델을 이용하여 추정되며 radial basis interpolation model의 입력으로 이용된다. 우리의 성능 평가에 따르면 이 프레임 워크는 대체로 0.5N 이하의 힘 에러를 수직, 수평방향으로 보여주었다.

두 번째 목적은 비균일한 과학 정보의 인지적으로 정확한 전달이다. 과학정보 내

부의 비균일 성은 포함된 정보를 사용자가 잘못 이해하도록 할수 있다. 예를 들어 AFM을 이용한 물체의 강성과 형태 정보를 전달할 때, 기존의 균일 강체를 전달하기 위해 개발된 햅틱 렌더링 알고리즘은 정확한 정보 전달에 실패할 수 있다. 최악의 경우, 낮은 강성을 가진 높은 위치를 높은 강성을 가진 낮은 위치 보다 낮게 인지할 수 있는 것이다. 이 문제는 사용자가 물체의 형태를 알고자 할 때 일정한 힘을 가하면서 만진다는 force constancy 이론을 통해 설명할 수 있다. 이 논문에서는 메쉬 형태로 표현한 물체의 형태과 강성을 전달하는 topography compensation algorithm for mesh를 제시하고 검증하였다. 이 알고리즘은 force constancy를 바탕으로 물체의 형태를 변형하여 사용자가 정확한 강성 정보와 형태 정보를 모두 느낄 수 있게 한다.. 또한 기존의 햅틱 렌더링 알고리즘의 렌더링 결과를 비교하고 사용자의 실험도 수행하여 제시한 알고리즘의 성능을 평가하였다. 그 결과 우리의 알고리즘은 물체의 형태를 더 잘 느끼게 하고 그에 걸리는 시간도 줄여 더 정확하고 효과적인 정보 지각화가 가능하게 하였다.

# 감 사 의 글

학부 시절부터 생각하면 포항에서 보낸 시간이 10년이 훌쩍 넘게 지냈습니다. 오랜 시간을 들여 성과를 이루었습니다만, 이는 모두의 도움이 있었던 덕분입니다.

긴 세월 뒷바라지를 해주신 부모님께 감사드립니다. 8년의 꽉 채운 기간 동안 부모님의 응원과 믿음이 많은 도움이 되었습니다. 대학원 생활을 지도해주신 최승문 교수님께 많은 감사를 드립니다. 교수님의 훌륭하신 지도가 아니었으면 현재의 성과는 있을 수 없었을 것입니다. 또한 대학원 생활의 후반부를 도와주신 석희 형에게도 감사를 드립니다. 좋은 선배이고 훌륭한 조언자가 되어 주셨습니다. 바쁜 시간을 쪼개어 논문 심사를 맞아주신 한성호 교수님, 정완균 교수님, 안희갑 교수님, 한보형 교수님께도 감사드립니다.

긴 대학원 생활 동안 많은 선배들의 도움을 받았습니다. 재인이 형, 성길이 형, 종현이 형, 그리고 진욱이 형에게 감사드립니다. 영어 문제로 많은 도움을 받은 인이에게도 고마운 마음이 가득합니다. 오랜 기간 동안 함께하고 즐겁게 보낸 인욱, 건혁, 종만, 호진, 성환, 그리고 Reza와 재봉이에게 감사의 마음을 보냅니다. 먼저 사회에 나간 동기와 후배들, 재훈, 재영, 채현, 갑종, 경표, 명찬과 보낸 시간도 즐거웠습니다. 수원에 있는 시간을 같이 한 Norman, 홍채, 윤아, Arsen, Waseem에게도 고마운 마음을 표합니다. 같이 있던 시간은 짧았지만 호준, 준석에게도 감사합니다. 대학원 생활의 초입에 있는 이 친구들에게 많은 성과가 앞에 있기를 바랍니다. 또한 언제나 다양하게 귀찮은 일을 가져간 김미자 선생님, 장혜자 선생님, 조동현 선생님, 그리고 오송이 씨에게도 감사드립니다.

많은 사람들 덕분에 긴 학업과 연구의 시간을 보낼 수 있었습니다. 모두의 앞날에 행복함이 가득하기를 바랍니다.

# Publications

## International Journals

1. **Sunghoon Yim**, Seokhee Jeon, and Seungmoon Choi, "Topography Compensation for Haptization of a Mesh Object and Its Stiffness Distribution," *IEEE Transactions on Haptics*, 2013 (Accepted, Early Access Article).

2. Hojin Lee, Gabjong Han, In Lee, **Sunghoon Yim**, Kyungpyo Hong, Hyeseon Lee, and Seungmoon Choi, "Haptic Assistance for Memorization of 2D Selection Sequences," *IEEE Transactions on Human-Machine Systems*, Vol. 43, No. 6, pp. 643-649, 2013.

3. Inwook Hwang, **Sunghoon Yim**, and Seungmoon Choi, "Haptic Discrimination of Virtual Surface Slope," *Virtual Reality*, Vol. 16, No. 3, pp. 205-218, 2013.

4. **Sunghoon Yim**, Sungkil Lee, and Seungmoon Choi, "Evaluation of Motion-Based Interaction for Mobile Devices: A Case Study on Image Browsing," *Interacting with Computers*, Vol. 23, No. 3, pp. 268-278, 2011.

5. Sangki Kim, Gunhyuk Park, **Sunghoon Yim**, Gabjong Han, Seokhee Jeon,

Seungmoon Choi, and Seungjin Choi, ”Gesture-Recognizing Hand-Held Interface with Vibrotactile Feedback for 3D Interaction,” *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 3, pp. 1169-1177, 2009.

6. Jane Hwang, Jaehoon Jung, **Sunghoon Yim**, Jaeyoung Cheon, Sungkil Lee, Seungmoon Choi, and Gerard J. Kim, “Requirements, Implementation and Applications of Hand-held Virtual Reality,” *International Journal of Virtual Reality*, Vol. 5, No. 2, pp. 59-66, 2006.

## International Conferences

1. **Sunghoon Yim**, Seokhee Jeon, and Seungmoon Choi, “Normal and Tangential Force Decomposition and Augmentation Based on Contact Centroid,” In *Asia Haptics*, 2014 (to be published, Honorable mention - Final candidate for the best demo award).

2. **Sunghoon Yim** and Seungmoon Choi, “Shape Modeling of Soft Real Objects Using Force-Feedback Haptic Interface,” In *Proceedings of the IEEE Haptics Symposium*, pp. 479-484, 2012.

3. Sunghwan Shin, In Lee, Hojin Lee, Gabjong Han, Kyungpyo Hong, **Sunghoon Yim**, Jongwon Lee, YoungJin Park, Byeong Ki Kang, Dae Ho Ryoo, Dae Whan Kim, Seungmoon Choi, and Wan Kyun Chung, “Haptic Simulation of Refrigerator Door,” In *Proceedings of the IEEE Haptics Symposium*, pp. 147-154, 2012 (Long oral presentation; Acceptance rate = 26; Candidate for the Best Paper Award).

4. **Sunghoon Yim**, Jane Hwang, Seungmoon Choi, and Gerard J. Kim, “Image

Browsing in Mobile Device Using User Motion Tracking," In *Proceedings of the International Symposium on Ubiquitous Virtual Reality*, 2007.

5. Hojin Lee, Gabjong Han, In Lee, **Sunghoon Yim**, Kyungpyo Hong, and Seungmoon Choi, "Effect of Active and Passive Haptic Sensory Information on Memory for 2D Sequential Selection Task," In *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR)*, pp. 52-54, 2011.

## International Conferences; Nonreferred Papers/Posters/Demonstrations

1. **Sunghoon Yim**, Seokhee Jeon, and Seungmoon Choi, "Haptic Augmentation for Palpation Using Contact Centroid." *International BioMedical Engineering Conference (IBEC)*, 2014.

2. **Sunghoon Yim**, Seokhee Jeon, and Seungmoon Choi, "Data-driven haptic modeling and rendering of frictional sliding contact with soft objects for medical training." *The International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2014.

3. **Sunghoon Yim** and Seungmoon Choi, "Shape Modeling of Soft Real Objects Using Force-Feedback Haptic Interface," Demonstrated in *IEEE Haptics Symposium*, 2012.

4. **Sunghoon Yim**, Seokhee Jeon, and Seungmoon Choi, "Stiffness Modulation of Inhomogeneous Real Objects for Haptic Augmented Reality Using Shape Models," *Korea-Japan Workshop on Mixed Reality (KJMR)*, 2012.

5. **Sunghoon Yim**, Seokhee Jeon, Seungmoon Choi, and Matthias Harders, "Progresses for haptic augmented reality," Demonstrated in *the World Haptics*

*Conference (WHC)*, 2011.

## Domestic Journals

1. Sangki Kim, Gunhyuk Park, Seokhee Jeon, **Sunghoon Yim**, Gabjong Han, Seungmoon Choi, and Seungjin Choi, "HMM-based Motion Recognition with 3-D Acceleration Signal," *Journal of th KIISE: Computing Practices and Letters*, vol. 15, no. 3, pp. 216-220, 2009.

2. **Sunghoon Yim**, Jane Hwang, Seungmoon Choi and Gerard J. Kim, "Image Browsing in Mobile Devices using Motion Tracking," *Journal of HCI Korea*, Vol. 3, No. 1, pp. 49-56, 2008 (invited).

## Domestic Conferences

1. **Sunghoon Yim** and Seungmoon Choi, "Modeling System Improvement for Haptic Augmented Reality," In *Proceedings of the 7th KROC*, pp. 251-254, 2012.

2. **Sunghoon Yim** and Seungmoon Choi, "Shape Modeling of Soft Real Objects Using Haptic Interface with Force Sensor," In *Proceedings of the HCI Korea*, pp. 3-5, 2012.

3. **Sunghoon Yim**, Seokhee Jeon and Seungmoon Choi, "Stiffness Modulation of Inhomogeneous Real Objects for Haptic Augmented Reality," In *Proceedings of the 6th KROC*, pp. 161-164, 2011.

4. Gunhyuk Park, Sangki Kim, **Sunghoon Yim**, Gabjong Han, Seungmoon Choi, Seungjin Choi, Hongjun Eoh and Sunyoung Cho, "Improved Motion-Recognizing

Remote Controller for Realistic Contents," In *Proceedings of the HCI Korea*, pp. 396-401, 2009.

5. **Sunghoon Yim** and Seungmoon Choi, "Design and Evaluation of Motion-based Interface for Image Browsing in Mobile Devices," In *Proceedings of the HCI Korea*, pp. 40-44, 2009 (a final candidate for the best paper award).

6. Sangki Kim, Gunhyuk Park, Seokhee Jeon, **Sunghoon Yim**, Gabjong Han, Seungmoon Choi, and Seungjin Choi, "HMM-based Motion Recognition with 3-D Acceleration Signal," In *Proceedings of the KIISE Fall Conference*, pp. 69-70, 2008 (Winner of the best paper award ).

7. **Sunghoon Yim**, Jane Hwang, Seungmoon Choi and Gerard J. Kim, "Image Browsing in Mobile Devices Using User Motion Tracking," In *Proceedings of the HCI Korea*, pp. 440-446, 2007.