DATA-DRIVEN HAPTIC MODELING AND RENDERING OF REALISTIC VIRTUAL TEXTURED SURFACES

Heather M. Culbertson

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2015

Katherine J. Kuchenbecker, Supervisor of Dissertation
Associate Professor, Department of Mechanical Engineering and Applied Mechanics

Prashant Purohit, Graduate Group Chairperson
Associate Professor, Department of Mechanical Engineering and Applied Mechanics

Dissertation Committee:
Mark Yim
Professor of Mechanical Engineering and Applied Mechanics
Katherine J. Kuchenbecker
Associate Professor of Mechanical Engineering and Applied Mechanics
Daniel D. Lee
Professor of Electrical and Systems Engineering
Yon Visell
Assistant Professor of Electrical and Computer Engineering, Drexel University

ACKNOWLEDGMENTS

Finishing this thesis would not have been possible without the love and support of so many people in my life. First I would like to thank my advisor, Katherine J. Kuchenbecker, for her encouragement and support over the last five years. I have been extremely lucky to work with her, and she has always pushed me to be the best researcher and engineer I can be. She has been an inspiration to me in her dedication and skill in both research and teaching, and I hope to one day to follow in her footsteps and become a professor.

It's been a honor to work will all of the members of the Penn Haptics Group. Thank you for always making me feel welcome in lab and making it such a fun place to work. Even though many lab members have come and gone over the years, know that each of you have touched my life and my research in some way. A special thank you to Joe Romano, Pablo Castillo, Ben Goodman, Tim Herrman, Raven Hooper, Nick Pesta, Ettie Unwin, Craig McDonald, Juanjo Lopez Delgado, and Shweta Krishnan who have worked with me on several projects over the years.

I would especially like to thank Rebecca Pierce for her support and encouragement over the last five years. Thank you for giving me countless opportunities to bounce ideas off of you, while simultaneously reminding me of when it is time to step away from the research and come back to it later with a clear head. Thank you for being such a caring and wonderful person, and for reminding me what is really important in life.

I also owe a great deal to my high school science teacher, Don Rotsma, who convinced me to take his physics class when I was hesitant that I could handle the topic. This class really started my love for engineering and mechanics, and without it my life trajectory would have been very different. Thank you for fostering in me such a passion for science that has really molded the rest of my life.

Finally I would like to thank my family, without whose love and support this thesis would not have been possible. Mom, thank you for supporting me in my decision to move across the country for graduate school and for never stopping to believe in me and my dreams. You have been my personal cheerleader for as long as I can remember and I could not have done this without your unflagging encouragement. Dad, thank you for fostering in me a love of building and creating as a child. Although I wish you were still here to share this moment with me, I know that you are watching with pride from heaven. Daniel, you are the best big brother I could have asked for. Thank you for always being there for me and sharing my love of all things nerdy.

The research presented in this thesis was generously supported by a grant from the National Science Foundation. My graduate studies were funded through research fellowships from the Department of Mechanical Engineering and Applied Mechanics at the University of Pennsylvania and the National Science Foundation Graduate Research Fellowship Program. These fellowships have given me invaluable freedom to pursue the research I am most passionate about.

ABSTRACT

DATA-DRIVEN HAPTIC MODELING AND RENDERING OF
REALISTIC VIRTUAL TEXTURED SURFACES

Heather M. Culbertson

Katherine J. Kuchenbecker

The haptic sensations one feels when interacting with physical objects create a rich and varied impression of the objects, allowing one to gather information about texture, shape, compressibility, and other physical characteristics. The human sense of touch excels at sensing and interpreting these haptic cues, even when the object is felt through an intermediary tool instead of directly with a bare finger. Dragging, pressing, and tapping a tool on the object allow you to sense the object's roughness, slipperiness, and hardness as a combination of vibrations and forces. Unfortunately, the richness of these interaction cues is missing from many virtual environments, leading to a less satisfying and less immersive experience than one encounters in the physical world. However, we can create the perceptual illusion of touching a real object by displaying the appropriate haptic signals during virtual interactions.

This thesis presents methods for creating haptic models of textured surfaces from acceleration, force, and speed data recorded during physical interactions. The models are then used to synthesize haptic signals that are displayed to the user during rendering through vibrotactile and/or kinesthetic feedback. The haptic signals, which are a function of the interaction conditions and motions used during rendering, must respond realistically to the user's motions in the virtual environment. We conducted

human subject studies to test how well our virtual surfaces capture the psychophysical dimensions humans perceive when exploring textured surfaces with a tool.

Three haptic rendering systems were created for displaying virtual surfaces using these surface models. An initial system displayed virtual versions of textured surfaces on a tablet computer using models of the texture vibrations induced when dragging a tool across the real surfaces. An evaluation of the system showed that displaying the texture vibrations accurately captured the surface's roughness, but additional modeling and rendering considerations were needed to capture the full feel of the surface. Using these results, a second system was created for rendering a more complete three-dimensional version of the haptic surfaces including surface friction and event-based tapping transients in addition to the texture vibrations. An evaluation of this system showed that we have created the most realistic haptic surfaces to date. The force-feedback haptic device used in this system, however, was not without its limitations, including low surface stiffness and undesired inertia and friction. We developed an ungrounded haptic augmented reality system to overcome these limitations. This system allowed us to change the perceived texture and friction of a physical three-dimensional object using the previously-developed haptic surface models.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The sense of touch is so fundamental to everyday human experiences that it is often overlooked. However, completing even simple tasks is challenging when your sense of touch is impaired. Recall trying to pick something up after your hand has fallen asleep. This everyday task becomes very difficult when you cannot feel the shape of the object or determine if you are applying enough force to keep the object from slipping. Now try picking up the same object without looking. Even without sight you can use your sense of touch to help you adjust your hand until you have a stable grip on the object. Feedback provided through the sense of touch makes such a simple task trivial for humans to successfully complete. However, modern virtual reality and human-computer interactions are often like the example where your hand falls asleep: tactile cues are completely removed, forcing you to complete tasks using only visual feedback.

We are increasingly interacting with the world through technology such as a computer, smartphone, or tablet. Traditionally the most common physical human-computer interactions have been with a keyboard or mouse, not with anything on the screen. Both finger- and stylus-based touchscreens have recently become more mainstream and allow us to directly manipulate objects on the screen, but they do not allow us to feel the objects with which we are interacting. Textured images on the screen feel as smooth as glass and leave the user with an unsatisfying experience. Adding the sense of touch back to these virtual interactions is necessary to create more immersive and usable devices.

The sensations one feels when interacting with the physical world are so rich and varied that fully capturing the depth of haptic information is a complex and multi-faceted problem. For example, if you pick up a tool such as a pen and gently drag its tip across the table, a rock, or the fabric of your shirt, you are able to feel variations in the textures and friction even though you are not directly touching the surface with your finger. Pressing the tool into the different surfaces provides information about surface stiffness. The human sense of touch excels at sensing and interpreting these haptic signals to gather information about the physical contacts taking place. Traditional haptic interfaces for virtual reality provide some of the information encoded in these signals, but they are unable to fully capture the full richness of natural interactions.

1.1 Human Tactile Sensing

Humans interact with the physical world through the use of our five senses: hearing, vision, smell, taste, and touch. Each of these senses provides us with a different type of sensory input, which we use to analyze and connect with the world around us. The first four senses listed provide information through localized sensory organs, our ears, eyes, nose, and tongue, respectively. The sense of touch, however, is not localized to a specific region of the body; instead it is distributed throughout the entire body through the touch sensory organ, our skin. The glabrous (nonhairy) skin of our hands is more sensitive to external touch stimuli than the nonglabrous (hairy) skin of our arms and legs [55]. Specialized cells embedded in our skin allow us to experience a wide array of haptic sensations including temperature, pain, vibration, pressure, and shear forces. Each of these cells is specialized to provide information about a specific component of our sense of touch. The *thermoreceptors* sense changes in skin temperature and provide information about the temperature of the surrounding environment and the thermal properties of objects that are being touched [43]. The *nociceptors* sense pain, which helps to protect us by reacting to potentially damaging stimuli [75]. The *mechanoreceptors* sense pressure, shear forces, slip, and vibration. There are several different types of mechanoreceptors, which each sense a component of the haptic sensations that are the focus of this thesis.

The mechanoreceptors are divided into two main groups, fast adapting and slowly adapting, which describe the temporal resolution of the mechanoreceptors [61]. Fast

Table 1.1: Mechanoreceptors in the human hand [9, 52, 53]

Mechanoreceptor	Response Rate	Response Type
Meissner Corpuscles (FA-I)	$5 - 50$ Hz	rate of skin deformation, slip
Merkel Disks (SA-I)	< 5 Hz	edges and spatial features
Pacinian Corpuscle (FA-II)	$20 - 1000$ Hz	acceleration of the skin
Ruffini Endings (SA-II)	< 5 Hz	skin stretch

adapting mechanoreceptors capture transient signals, and slowly adapting mechanoreceptors capture static stimuli. They are further designated as Type I and Type II. Type I mechanoreceptors (Meissner corpuscles and Merkel disks) have small receptive fields, and Type II mechanoreceptors (Pacinian corpuscles and Ruffini endings) have large receptive fields [61]. The characteristics of the four mechanoreceptors are summarized in Table 1.1.

Meissner corpuscles (FA-I) are sensitive to low-frequency vibrations and sense the rate of skin deformation [52]. This skin deformation is often due to the minuscule slipping of a grasped object. The Meissner corpuscles sense this slip and signal that a higher grip force is needed, which helps ensure successful grasping and lifting of an object [54]. Merkel disks (SA-I) detect edges and spatial features; they have been shown to play the largest role in sensing texture and roughness of objects felt with a bare finger [54]. Although they can sense surface features such as a Braille dot during either static touch or lateral scanning with a bare finger, the sensations felt during scanning are more salient. Pacinian corpuscles (FA-II) sense a wide range of high-frequency vibrations that are present during transient contacts with objects (e.g.,

collisions or dragging across a bumpy surface), especially during contacts through an intermediary tool [54]. However, like Meissner corpuscles, Pacinian corpuscles are responsive only to changing signals. Both FA mechanoreceptors fire when contact is initially made, but they stop firing during sustained contact [54]. Ruffini endings (SA-II) sense skin stretch [52], which partially allows for the perception of the direction of object motion or force [53]. The Ruffini endings can also sense the skin stretch over tendons, which can also provide proprioceptive information of hand location or pose [53].

When you interact with and touch objects in your environment, however, you do not recognize the individual haptic signals that your body is sensing. Instead your brain attributes the sensations to the object or world you are touching through the principle of distal attribution [74]. This principle can be applied to touching physical objects, touching remote objects through teleoperation, and touching virtual objects. In order to achieve full distal attribution, the sensory input must be authentically related to the motions made and the mode of touch. To create highly realistic virtual objects, the haptic signals that are displayed to the user must mimic the signals experienced during interactions with physical objects. The first step to achieving this realism is to determine what haptic signals are sensed for a specific object and mode of touch.

1.2 Touch Modalities

This thesis focuses on haptics as it pertains to the human hand. The glabrous (non-hairy) skin of the hands has the highest concentration of mechanoreceptors in the human body [55]. Your hands are adept at doing a wide array of tasks including playing music, threading a needle, juggling, and buttoning a shirt. Although the first thing many people think of concerning their sense of touch involves manipulating and feeling objects through their bare fingers, there are also a wide array of important haptic interactions that are accomplished through the use of a tool: a student taking notes on paper, an artist painting a new masterpiece, a surgeon finely cutting or suturing tissue, or a paleontologist carefully excavating a new fossil. Both types of haptic interactions, with bare fingers or through a tool, are essential to everyday life. However, the types of haptic cues that are experienced in the two instances are different.

For example, when you are touching an object with your bare fingers, you engage in many different exploratory procedures to gather information about an object [69]. If you drag your finger across a textured surface, you are provided with a spatial picture of the surface features, as well as an estimation of the friction through the lateral forces. By pressing your finger into the surface, you can gauge its stiffness. If you rest your finger on the surface, you can estimate its temperature and thermal properties. These sensations combined make up your perception of that object and can be divided into four perceptual dimensions: roughness (both micro and macro),

slipperiness, hardness, and warmness [81].

In contrast, touching objects with a tool is a temporal interaction. Spatial cues are not present because the deformation of the fingers is due to the shape of the pen and not the surface features of the object [59]. Spatial information cannot be sensed with a single point of contact. Instead motion of the tool is required to receive useful haptic information. When you drag a tool across a surface, vibrations are created when the tooltip hits the surface features [67]. These vibrations provide information about the roughness of the surface [71, 119]. If you tap or press the tool into the surface, you can estimate its stiffness [65]. The sensations felt with the tool are dependent on the tool's stiffness, size, and shape. Although touching an object with a tool allows us to learn a lot about how the object feels, there is some information that is lost compared to touching the object with a bare finger, most notably temperature and macro roughness. Consequently, the perceptual dimensions for tool-mediated interactions are roughness (micro), slipperiness, and hardness [119].

1.3 Haptic Interfaces

After the desired haptic signals are chosen, they must be displayed to a user. Haptic interfaces are systems that are designed to display haptic signals to a user in response to events in a physical or virtual world. These interfaces can take many different forms and are commonly either attached to the body (wearable haptics) or are grounded devices that a human grasps. Interfaces have been designed to display

a wide array of haptic signals and are typically categorized by their ability to display either kinesthetic or tactile information.

Kinesthetic haptic devices display forces through a tool or handle that is grasped by the user. A large number of these kinesthetic devices are grounded lightweight robotic arms that sit on a tabletop (or the floor for larger devices) and transmit force from the ground to the user's hand. The user grasps an actuated handle to produce motion, which is commonly measured through encoders. There are two main types of kinesthetic haptic devices, impedance-type devices and admittance-type devices, which differ in the relationship between user motion and output force. With impedance-type devices, the output force is computed as a function of position. For example, if the user was exploring a virtual environment, a force would be applied to stop the user from moving when they contact a virtual object or wall. No force is displayed when the user is exploring free space, which is necessary for a satisfying virtual experience [76]. With admittance-type devices, the desired position is computed as a function of force. These devices measure the force that the user is exerting and allow a corresponding motion. However, admittance-type devices have inherent limitations that prevent the user from moving freely in free space. This thesis focuses on force feedback provided by impedance-type haptic devices.

Tactile haptic devices interact directly with the skin and can display vibrations, local shape, contact location, slip, and temperature [38]. There is a large body of prior work that has explored the use of tactile actuators such as pin arrays (e.g., [56, 109])

and balloon arrays (e.g., [57]) to display spatial information to the user's fingertip, as well as tactile actuators to display temperature (e.g., [44]), slip (e.g., [98, 103, 111]), and contact location (e.g., [92]). This thesis focuses on vibration as tactile feedback because of its prevalence in tool-mediated interactions.

Consumers might be most familiar with vibrotactile feedback though vibration alerts in cell phones or vibrations in game controllers to signify events in a game. The common type of vibrotactile actuator that is embedded in phones is an eccentric mass motor that has a mass attached a distance from the shaft of the motor. When the motor is turned on, the device experiences steady vibrations. These motors are small, lightweight, and inexpensive, but the quality of the haptic cues they provide leaves many things to be desired. When controlling such motors you can specify the amplitude or the frequency of the vibrations, not both. Therefore, the quality of the vibrations that are output are not sufficient for displaying the rich vibrations experienced during interactions with the physical world. A second type of vibrotactile actuator is the linear resonant actuator (LRA), which consist of a mass on a spring in a magnetic field. They have successfully been used to display vibrations to mimic button clicks [64]. As their name suggests, however, the LRAs have a single resonant frequency. Similar to the eccentric mass motors, the LRAs do not provide rich vibrotactile feedback because of their narrow bandwidth. To display vibrations to match the richness of tactile cues with the physical world, many researchers have turned to voicecoil actuators, which are stronger and have a higher bandwidth than LRAs.

These actuators work much the same way as a speaker, except they output vibrations instead of sound. They are a mass-spring-damper system composed of a high-mass magnet that is suspended between two flexure springs in an electromagnetic coil. The desired vibration signal is commanded to the coil, which creates an electromagnetic field to either attract or repel the magnet. Since the magnet has a large mass compared to the rest of the actuator, its movement inside the coil creates large vibrations that are transmitted to whatever the actuator is attached to.

1.4 Thesis Overview and Contributions

The goal of this thesis is to leverage our knowledge of the human sense of touch to create virtual haptic interactions that are indistinguishable from their real counterparts. I take an approach that has been coined "haptography" (haptic photography) [63] because it allows a person to record the feel of an interesting interaction in much the same way that traditional photography allows a person to visually record an interesting scene or object. The underlying hypothesis is that modeling and rendering the haptic signals generated when interacting with physical surfaces through a tool will allow us to create realistic virtual textured surfaces. I present detailed methods for creating these haptic texture models (Chapter 2). Then I describe and analyze three systems for rendering the virtual surfaces: two-dimensional tablet (Chapter 3), three-dimensional virtual reality (Chapter 4), and three-dimensional haptic augmented reality (Chapter 5). I introduce our open-source Penn Haptic Texture Toolkit, which

includes texture data and rendering methods (Appendix A). The primary contributions of this thesis are:

- Chapter 2: Recording and Modeling of Haptic Textures
  - An efficient method for creating haptic texture models from brief tool acceleration, force, and speed signals that are recorded in an unconstrained manner. Evaluation of this method showed that ten-second data recordings are sufficient for capturing the behavior of the texture-tool system over a wide range of interaction conditions. Although the texture models' spectral properties were sensitive to differences in individual hand dynamics, the effect was small compared to the changes in spectral properties due to normal force and scanning speed.
  - Methods for stably generating haptic texture vibrations in real time using measurements of user normal force and scanning speed. A new model interpolation scheme was implemented to ensure stability.
- Chapter 3: Rendering and Evaluation of Haptic Textures
  - Methods for rendering haptic texture vibrations using a two-dimensional tablet system.
  - Analysis through a human-subject study indicated that displaying the texture vibrations accurately captured the texture's roughness information. However, the texture's hardness information was not fully captured by

the texture vibrations and the texture's slipperiness information was not captured at all by the texture vibrations.

- Chapter 4: Three-Dimensional Rendering of Haptic Surfaces
  - Evidence that texture vibrations must respond to the user's speed, but not necessarily to the user's force, in order to achieve realism.
  - Methods for rendering three-dimensional haptic surfaces through a combination of friction, tapping transient, and texture vibration models.
  - Analysis indicated that the combination of the three modes of feedback accurately captures the real surfaces' roughness, hardness, and slipperiness. However, not all surfaces require that the three model components be displayed for realism.
- Chapter 5: Displaying Texture and Friction via Haptic Augmented Reality
  - Methods for determining the contact position on a physical three-dimensional object using a geometric model of the object and measurements of the applied force and torque.
  - A novel haptic augmented reality device for augmenting the perceived texture and friction of a physical three-dimensional object.
- Appendix A: The Penn Haptic Texture Toolkit
  - Creation of an open-source database of texture data, models, and rendering

code for 100 surfaces. Researchers can use the texture data to test their own modeling schemes and directly compare to our texture models. The toolkit has been downloaded over 500 times.

Chapter 2

Recording and Modeling of Haptic Textures

Dragging a tool across a textured object elicits a range of haptic cues that allow you to form a perceptual impression of the object even though your finger does not directly contact it. For example, if you pick up a pen and drag it across a table, a rock, or a swatch of fabric, you can feel variations in the texture even though your finger is not directly touching the surface. The vibrations that are felt during dragging are caused by contact between the tooltip and small surface features [67] and are transmitted to your hand through the rigid link of the tool. The human sense of touch excels at sensing and interpreting these vibrations to gather information about the physical world [59]. Unfortunately, the richness of these interaction cues is missing from many virtual environments, leading to a less satisfying and less immersive experience than

one encounters in the physical world.

Feeling appropriate tool vibrations during virtual interactions can create the perceptual illusion that one is touching a real surface. Following the principle of distal attribution, for virtual surfaces to be realistic they must convince you that the vibrations come from the contact between the tool and the virtual surface by behaving in a physically appropriate manner [74]. However, most modern haptic algorithms and devices are incapable of outputting high-fidelity reproductions of such vibrations [17, 62]. Many approaches to representing textures have been proposed, but no consensus exists [85]. However, measurement-based models can be significantly more realistic than models that are simply hand-tuned without a basis in physical reality [83]. Recent research has proven the technological feasibility of creating texture models from recorded high-frequency tool acceleration signals [24, 95, 96].

This Chapter presents a new method for the creation of haptic texture models from short acceleration, force, and speed signals recorded when an individual explores a textured surface with an instrumented tool in a natural and unconstrained manner. This method is efficient in both required data and computation time. We focus particularly on the vibrations caused by tool-surface contact; these signals relate to, but do not completely capture the hardness and slipperiness of the surface. Thus, our data-driven approach represents a carefully chosen practical simplification of the tool-surface interaction. This Chapter presents work published in the Proceedings of the IEEE World Haptics Conference [25] and the IEEE Transactions on Haptics [26].

2.1 Prior Work in Texture Modeling

Modeling and rendering realistic tool-mediated textures is a complex and multi-faceted problem that has not been fully solved. One of the limitations to finding a solution to this problem is the inherent complexity of interactions between a tool and the surface. Complete physics-based simulations are too computationally complex for real-time rendering [85]. A physics-based simulation that was recently created to explain the transients caused by a tool contacting a textured surface showed a promising match to data recorded from a physical surface [77]. However, it could not be completed in real time and would thus not be applicable to rendering virtual textures.

To lower the complexity to a point where real-time execution is feasible, haptic texture rendering schemes seek to model textures via a simplification of the underlying physics. Many attempts have been made to recreate these interactions, but most have not been able to completely match the richness and usefulness of haptic feedback experienced during interactions with the physical world. One early approach to modeling textures involved the implementation of virtual springs to perturb the height of the user's hand [80]. This rendering approach was also applied to generate stochastic textures using Gaussian-distributed force fields [34]. However, the goal of these two rendering approaches was to produce distinguishable textures and not textures with a real-world basis.

An early approach that did have a basis in real textures analyzed the pixels of a

grayscale image to create a surface height profile that directly depended on shading value [6]. Data-driven modeling schemes such as this have an advantage over physics-based modeling in that they do not require tedious and potentially inaccurate hand-tuning. Rather, data-driven models seek to capture the output response of a system (e.g., force, acceleration) given user inputs (e.g., position, velocity, force) [83].

Several researchers have explored the use of data captured from real interactions to increase the realism of virtual haptic interactions. Past applications include the use of real-world data to virtually represent the sensations felt when cutting [84] and tapping [62]. Data-driven modeling has also recently been used in haptic augmented reality, which seeks to modulate the haptic properties of real objects using virtual feedback [40, 50].

Several approaches have applied the principles of data-driven modeling in the area of haptic textures. Direct playback of recorded vibrations, the simplest approach, has been implemented using voice-coil actuators [102] and a cable-driven system [97]; however, there are many drawbacks to this approach. Because the power and frequency content of the vibration must change as a function of the user's force and speed, one needs a method for interpolating between signals recorded under different conditions. One could simply always play the recording closest to the user's current force and speed, but switching between recordings when force and speed change will create perceptually noticeable artifacts in the new acceleration signal. Another possible method could be to output a weighted average of recorded acceleration sig-

nals as a function of force and speed. However, this summation in the time domain causes significant constructive and destructive interference between the signals and does not preserve frequency content. Furthermore, recordings require a large amount of storage space, and any peculiarities that happened to be captured would likely be perceptually noticeable when repeated in a loop. These pitfalls in directly playing recorded acceleration signals have motivated a range of work in creating data-driven texture models.

For example, Okamura et al. [82] represented patterned textures as a set of data-driven decaying sinusoids that depend on the user's speed and applied force. The vibrations used to represent the textures were superimposed on forces used to represent the stiffness of the surface for output to a force-feedback joystick. Guruswamy et al. [36] took a similar approach and created texture models based on a spatial distribution of infinite-impulse-response filters that are fit with decaying sinusoids.

Other researchers have explored the use of alternative types of data captured from real interactions to create virtual haptic texture models to increase realism. Pai et al. [86] modeled measured friction variations as an autoregressive process under the assumptions that the roughness of a surface is isotropic and people are sensitive only to the statistical features of varying friction. Vasudevan and Manivannan [107] used a SensAble PHANToM to drag across a textured surface and modeled the resulting haptic texture from the frequency spectrum of the tooltip's vertical displacements. Although normal force was kept constant, they allowed for variations in scanning

speed by interpolating between two texture datasets at different speeds.

From the many types of data recorded from interactions with physical surfaces, the use of tool vibrations has resulted in the most realistic haptic textures to date as rated by naive subjects [95]. The goal of the research presented in this Chapter is to build on the methods for creating data-driven texture models presented in [24, 95, 96] where the experimenter recorded data in a controlled manner by dragging the tool in a circular motion, keeping force and speed approximately constant. This method of recording data was difficult and tedious, and it resulted in interaction data that inevitably drifted in force and speed. Using a robot to record data might allow for more consistent force and speed. However, the texture models we create in this Chapter are a function of the entire tool-hand-surface system. Therefore, the dynamics of the robot would need to match the dynamics of the human hand-tool system to create accurate texture models, a challenging task. Furthermore, the accelerometer would pick up the vibrations of the robot itself, which would lead to inaccurate acceleration measurements. The ability to model data that is recorded without constraints on the experimenter's force and speed strongly motivated this work, so anyone could perform the data collection.

2.2 Recording the Feel of a Texture

The first step in creating a haptic texture model is to record data from tool interactions with real textures. This Section details the hardware and calculations needed

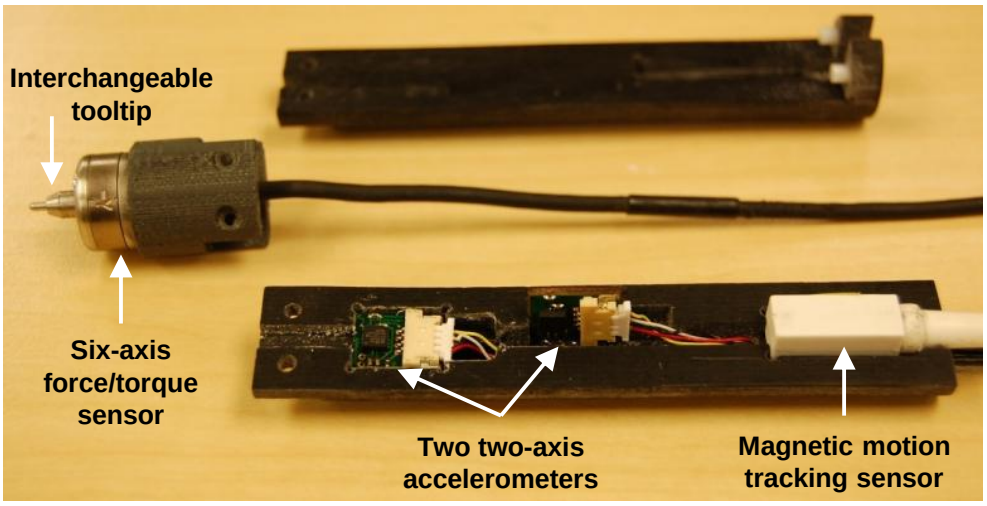

Figure 2.1: Haptic recording device. The tool measures position, orientation, force, and acceleration for three axes during interaction with texture samples.

to record and process the interaction data.

2.2.1 Data Collection System

We capture tool-surface interaction data using the custom haptic recording device first presented in [25] and shown in Fig. 2.1. This device records the tool's position, orientation, contact force, and high-frequency acceleration while it is dragged across a textured surface. A 16-bit USB analog-to-digital DAQ device from National Instruments records the force and acceleration data at a rate of 10 kHz. A custom Matlab GUI was created to interface with the recording system.

Magnetic Tracking Sensor

An 8 mm Ascension 3D Guidance TrakSTAR magnetic tracking sensor embedded in the back of the tool measures the tool's position and orientation at 240 Hz with a

resolution of 0.5 mm and 0.1 degrees respectively.

Force Sensor

The force sensor, which is located between the body of the tool and the tooltip, is a Nano17 SI-25-0.25 six-axis force/torque transducer from ATI Industrial Automation Inc. The force sensor can be fitted with a variety of tooltips. The tooltip used in this research was a stainless steel hemispherical tip with a diameter of 1.6 mm.

Accelerometers

Numerous three-axis analog accelerometers were considered to measure the tool vibrations, but none fit the required size, range, and bandwidth specifications. Consequently two high-bandwidth two-axis ± 18 g ADXL321 analog accelerometers were chosen, having a resonant frequency of 5.5 kHz and a noise level of 0.126 m/s^2. These accelerometers are rigidly embedded in the tool at right angles to allow for measurement of three orthogonal axes plus a fourth redundant axis. The accelerometers are band-limited by a five-pole low-pass Bessel filter at 1000 Hz to remove the effects of sensor resonance and frequencies above the bandwidth of human vibration detection [9]. Although humans cannot feel these higher frequencies, they can affect the feel of the surface because the channels that process auditory and touch frequency contents are linked [116]. Phase distortion, which is an artifact of the modeling approach taken in this work, does not directly affect the feel of textures because humans cannot discriminate phase shifts in haptic signals [18]. However, it could still interfere

with the tactile perception of the surface because it alters the way that synthesized vibrations sound.

2.2.2 Data Collection Procedure

Data is recorded while the user holds the haptic recording device and explores a selected surface using natural and unconstrained motions. The user is free to vary normal force and scanning speed during each ten-second recording; however, a range of normal forces and scanning speeds should be used to best capture the feel of the surface under different contact conditions. A single ten-second recording was made for each material presented.

2.2.3 Data Processing

The steps required to transform the recorded data into data usable for modeling are shown in Fig. 2.2. The three acceleration signals are first converted from volts to m/s^2. A high-pass filter at 20 Hz is applied to the three signals to remove gravity and the effects of purposeful human motion. Since humans cannot discriminate the direction of high-frequency vibrations [9], the three filtered signals are mapped onto a single axis using the DFT321 algorithm, which preserves the spectral and temporal properties of the three-axis signals [66].

The force signals are separated into the three force axes, and a constant rotation matrix is applied to rotate these force signals to the tool frame. The Euler angles

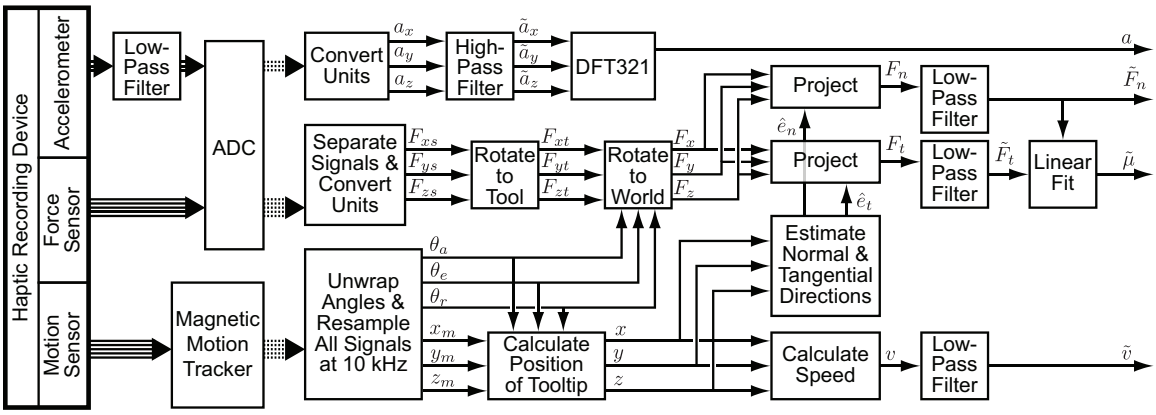

Figure 2.2: A diagram of our data recording and processing system. Acceleration and force are recorded at 10 kHz. Position and orientation are recorded at 240 Hz and upsampled to 10 kHz. A rotation matrix calculated from the orientation data at each time step is used to find the force vector and tooltip position vector in the world frame. The normal and tangential forces and tooltip speed are calculated and low-pass filtered.

and the positions are upsampled to 10 kHz to be at the same rate as the force measurements. A rotation matrix from the tool frame to the world frame of the magnetic tracker's field generator is calculated at each time step. The forces are rotated to the world frame, and the tooltip position in the world frame is found by adding the sensor position in the world frame to a rotated constant offset vector from the tracker sensor to the tooltip (obtained through calibration).

After recording, a plane is fit to the tooltip positions in the world frame to approximate the surface of the material and the unit vector normal to the surface plane ($\hat{e}_n$) is determined. The unit vector in the direction of motion ($\hat{e}_t$) is calculated at each time step. The inner product is calculated between $\hat{e}_n$ and the force vector in the world frame $\vec{F}$ to find the normal force (F_n), and separately the inner product is calculated between $\hat{e}_t$ and $\vec{F}$ to find the tangential force (F_t). F_n and F_t are used to calculate the friction coefficient for the surface by fitting a line to the plot using

Figure 2.3: Six textures that were haptically modeled with the presented methods. These materials were chosen to provide a set of textures that was diverse in roughness, hardness, slipperiness, and fineness of surface features.

a robust fit method in which outliers were removed. The slope of the line is the kinetic friction coefficient. The tooltip speed v is calculated as the magnitude of the discrete-time derivative of the tooltip position vector. The normal force, tangential force, and speed are low-pass filtered at 100 Hz to reduce noise.

2.2.4 Texture Samples

The recording system was used to characterize the six surfaces shown in Fig. 2.3. They were chosen to create a set that is diverse in roughness, hardness, slipperiness, and fineness of surface features. All materials were cut to 10.2 cm squares. The cloth materials (canvas, silk, vinyl) were mounted on a 0.32 cm thick piece of acrylic using double-sided tape. Although a limited set of materials was chosen for presentation in this work, the modeling and rendering methods are valid for a wide range of isotropic

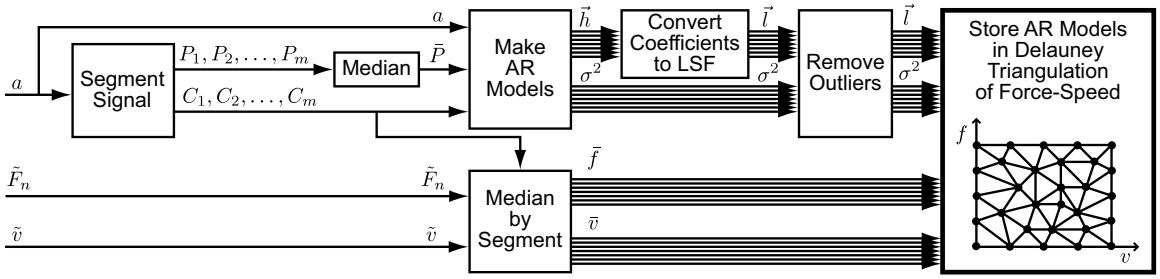

Figure 2.4: A diagram of our texture modeling system. The acceleration signal is fed into the segmenting algorithm, which returns the segmentation points and proposed AR order. An AR model is created for each segment and labeled with the median force and median speed during that segment. The coefficients are converted to LSFs to ensure stability during rendering. Outlier AR models are determined and rejected as appropriate. The remaining AR models are stored in a Delaunay triangulation in force-speed space.

and homogeneous materials, as demonstrated in [23].

2.3 Texture Modeling

This section details our method for creating haptic texture models from short acceleration, force, and speed signals that are recorded as described above. First we discuss the time series model used to represent the texture data. Then we describe the segmenting method used to parse the data into short stationary segments. Finally, we describe how the models are stored for later use in rendering virtual haptic textures. Fig. 2.4 shows a full diagram of the texture modeling process.

2.3.1 Model Structure

In [24], we used auto-regressive moving-average (ARMA) models. This type of model is typically used for weakly stationary data in which the signal evolves over time [16].

ARMA models were needed because the data signal was long (5 seconds), and because the force and speed were kept only approximately constant, which resulted in varying signal power and frequency content. However, if the assumption of strong stationarity can be made, the simpler autoregressive (AR) model structure is appropriate, as used in some other previous work [95, 96]. An AR model is an all-pole infinite impulse response (IIR) filter in which the system's next output is modeled as a linear combination of its previous outputs. The AR model order, p, specifies the number of previous output values used. The AR model structure is:

$$A(p)\, y(t) = e(t) + u(t) \tag{2.3.1}$$

where $A(p)$ is the array of AR coefficients, and $y(t)$, $e(t)$, and $u(t)$ are respectively the output, white-noise disturbance value, and residual at discrete time t. The discrete-time transfer function of an AR model is:

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} A_k z^{-k}} \tag{2.3.2}$$

The coefficients of each AR model are computed by minimizing the residual error variance using the Levinson-Durbin algorithm.

2.3.2 Data Segmentation

The recorded acceleration signals are not stationary because their power and frequency content directly depend on the unconstrained movement of the tool. However, the use of the AR model structure requires the assumption of strong stationarity. To

make this assumption valid, we break the acceleration signal down into a sequence of approximately stationary segments to model the signal as a piecewise autoregressive process. An individual AR model is then created for each segment. For each material, we segment the recorded acceleration signal using the Auto-PARM algorithm presented in [28]. This algorithm determines the segmentation breakpoints and the AR order of each segment by optimizing the minimum description length (MDL) objective function:

$$\begin{aligned} \text{MDL} = \log(m) + (m+1)\log(n) + \sum_{j=1}^{m+1} \log(p_j) \\ + \sum_{j=1}^{m+1} \frac{p_j + 2}{2} \log(n_j) + \sum_{j=1}^{m+1} \frac{n_j}{2} \log(2\pi\sigma_j^2) \end{aligned} \tag{2.3.3}$$

where m is the number of breakpoints, n is the length of data, and p_j, n_j, and σ_j^2 are respectively the AR order, length, and variance of the jth segment. The MDL function seeks to maximally compress the data by finding the best combination of the number of segments, the number of AR coefficients, and the resulting error. This maximum compression seeks to avoid the problem of oversegmenting the acceleration signal; oversegmenting may leave insufficient data to make a good AR model or may create redundant models. It also avoids the problem of overfitting individual AR models because although a model with more coefficients may fit the recorded data more precisely, it will be poor at prediction because it also fits the noise and random effects present in the data. A minimum segment span constraint of 50 datapoints was enforced to further prevent oversegmentation.

The recorded acceleration signal was fed into the Auto-PARM algorithm, which returned the location of the segment breakpoints and a proposed AR order for each segment. The number of AR coefficients was allowed to vary from one to twenty with equal probability. Auto-PARM implements an island-model genetic algorithm with 25 islands, which allows parallel computation to speed up convergence. A mutation operation prevents the algorithm from being trapped in local optima. The resulting segmentation for the acceleration signal recorded from canvas is shown in Fig. 2.5. Note that the segment boundaries often align with changes in either the force or speed signal even though only the acceleration signal was used in the segmentation process. For the dataset shown, the algorithm resulted in 19 segments with a mean length of 5230 datapoints (523 ms).

2.3.3 Creation of Texture Model Set from Segments

Once the breakpoints and AR orders for a material's dataset are returned by the segmenting algorithm, a texture model is created for that material as a set of autoregressive models. All AR models for a single material must have the same number of coefficients due to the need for interpolation between the AR models, as discussed later. We choose the median of the AR orders selected for all segments for that material, which for the data shown in Fig. 2.5 was 19 coefficients. We refit each segment with this median AR order and determine the segment's coefficients and variance. The AR coefficients are converted to line spectral frequencies (LSF) to ensure stabil-

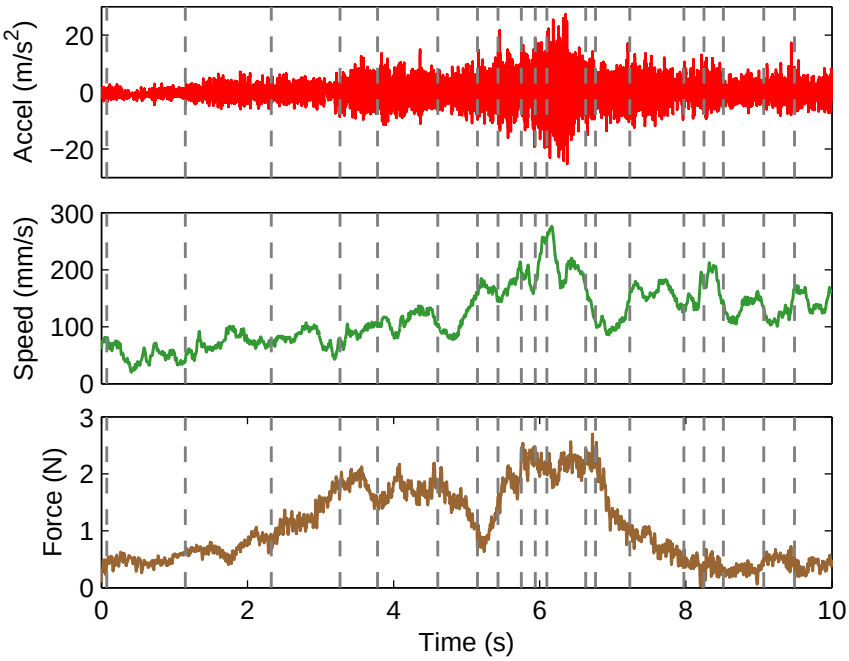

Figure 2.5: Recorded acceleration signal that was segmented using the AutoPARM algorithm. Dotted lines represent the segment breakpoints. Force and speed were not used during the segmenting process, but they are shown to illustrate the motions the subject used during data capture.

ity during rendering, as discussed in Section 2.4.1. Each AR model is associated with the median force and speed of the segment from which it was constructed.

Although [113] found that textures can be represented as a function of distance traveled when touched with a bare finger, we found that this approach was not applicable when using a tool because the dynamics of the tool-hand-surface system drastically changes as force and speed vary. For example, the tooltip may glide across the surface features when speed is increased [71], which alters both the power and spectral content of the resulting tool vibrations. Similarly, an increase in normal force requires the human to increase grip force, which increases the rigidity of the system and changes its vibrational characteristics. Thus, this work focuses on force

and speed as separate inputs to the system.

Due to the unconstrained method by which data is recorded, there is the chance that outlier AR models could exist if either force or speed varied greatly during a segment or if the tool or surface slipped during recording. Therefore, it is necessary to check for the existence of outlier AR models and remove them from the texture model set if appropriate. The AR models are placed in force-speed space and are compared to surrounding AR models on the basis of five criteria: the variance of error of the AR model and the power, spectral centroid, skewness, and kurtosis of the original segment. The expected behavior of the system is that these five values should increase monotonically as either force or speed increases. So a plane is fit to each of the five values versus force and speed. Potential outlier AR models are determined by the confidence intervals of the studentized residuals' t-distributions. If an AR model is determined to be an outlier under two or more criteria, the AR model is removed from the texture model set.

2.3.4 Model Storage

In prior work, the model forces and speeds were linearly spaced at set values and stored in a lookup table [24, 95, 96]. However, the segmentation and modeling process described above yields a set of AR models that are scattered in force-speed space. We thus construct a Delaunay triangulation using the z-score of the AR model forces and speeds, as shown in Fig. 2.6. Each vertex of a triangle represents the location of

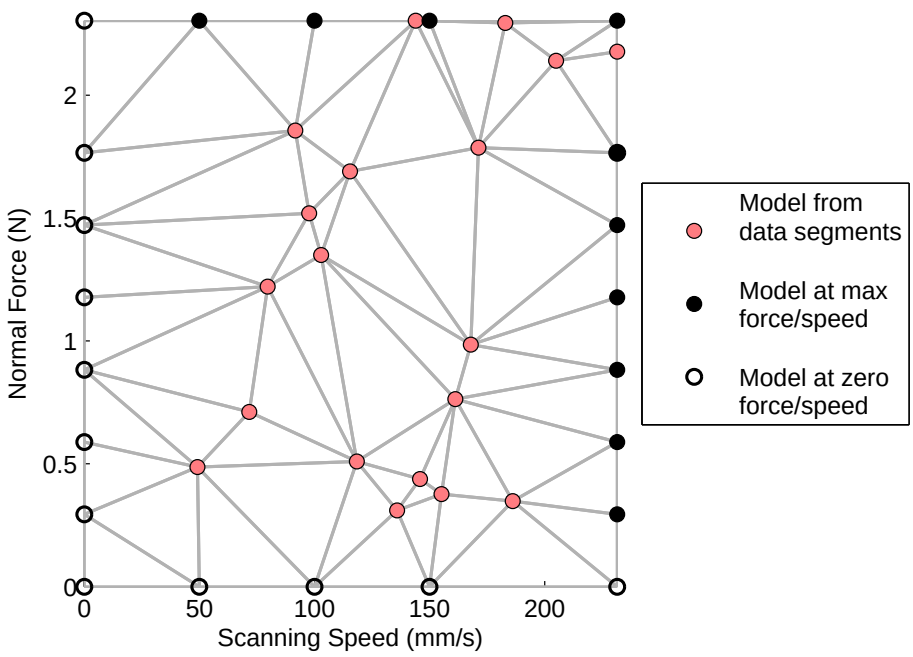

Figure 2.6: Delaunay triangulation for the canvas model set. Each vertex represents an AR model of a vibration segment recorded at that combination of force and speed.

an AR model in force-speed space.

The triangulation is extended to form a rectangular convex hull by adding linearly spaced AR models at maximum modeled force, maximum modeled speed, zero force, and zero speed. This step, which is done to simplify the rendering process, is accomplished by copying the coefficients of the closest AR model. For AR models at the maximum modeled force or speed, the variance is copied as well. For AR models at zero force or zero speed, the variance is set to zero so that no vibrations are generated if the user is not touching the surface or is stationary.

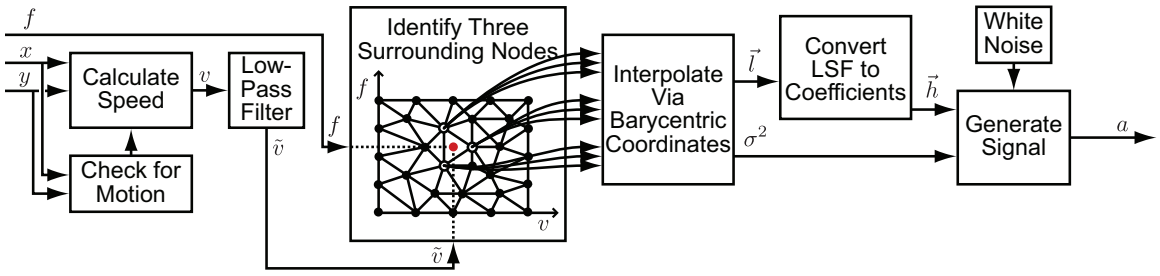

Figure 2.7: A diagram of our texture signal synthesis process using measurements of a user's current position and force. The position data is differentiated and low-pass filtered to obtain the speed. A collision detection algorithm is run to determine where the user's force and speed lie in the Delaunay triangulation. Barycentric coordinates are used to interpolate between the LSFs and variance of the three AR models at the vertices of the triangle. The interpolated LSFs are transformed to AR coefficients. The coefficient and variance values are passed to our LPC signal generator, which creates an appropriate acceleration waveform.

2.4 Generating Vibration Signals

This section describes how a model set created using the methods described above is used to synthesize texture vibration signals using only force and speed data. Fig. 2.7 shows a full diagram of the vibration signal synthesis process.

2.4.1 AR Model Interpolation

A sample force-speed trajectory is shown in Fig. 2.8. The force and speed are recorded at rates of 240 Hz and 10 kHz, respectively, and the force data is downsampled to 240 Hz. These force and speed values are used to determine which AR models to use in generating the vibration signal for display to the user. Simply choosing the AR model closest to the user's current force and speed creates the potential for high-frequency model switching, which can cause instability [73]. Similarly, directly interpolating the AR models' coefficients can move the poles of the resulting transfer function outside

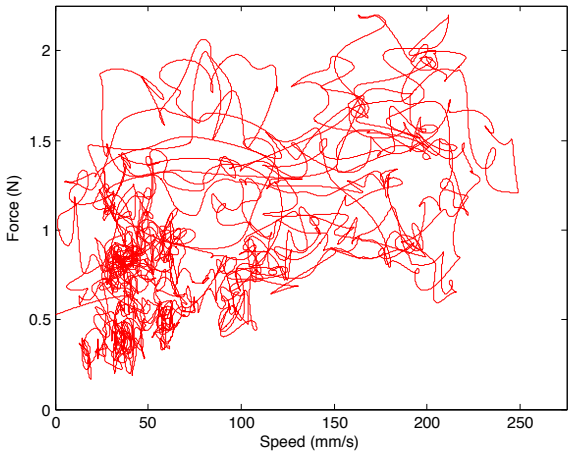

Figure 2.8: Sample force-speed trajectory used during synthesis of vibrations. This trajectory spans the range of forces and speeds typically used by individuals when interacting with textures through a tool.

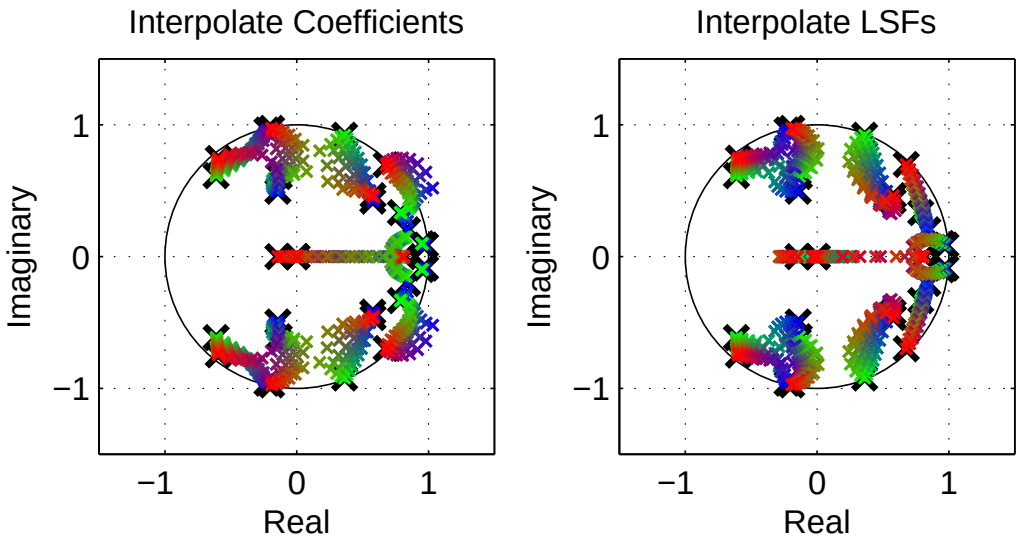

Figure 2.9: Loci of poles for interpolating between three models using (a) coefficients and (b) line spectral frequencies. When interpolating with the coefficients of the models, some poles travel outside the unit circle, resulting in a new unstable model. The line spectral frequencies ensure the stability of the model by requiring all poles to lie within the unit circle.

the unit circle, making the transfer function unstable, as shown in Fig. 2.9. Therefore, to ensure a stable system, one must convert each AR model's coefficients into line spectral frequencies (LSF) before interpolation. The LSFs are found by mapping the AR model's poles in the discrete plane onto the unit circle and are defined as the angles these mapped poles make with the real axis. This pole mapping ensures the poles remain inside the unit circle, as shown in Fig. 2.9, which is a necessary condition for stability.

We use a visibility-walk collision-detection algorithm [30] on the Delaunay triangulation to determine which triangle contains the user's current force and speed point. The user's force and speed are saturated at the maximum modeled force and maximum modeled speed to ensure that the force-speed trajectory remains inside the convex hull of the triangulation. Barycentric coordinates are used to provide a weighted average of the LSFs and variance values of the three AR models at the triangle vertices. These weights are found using:

$$\begin{aligned}
\lambda_1 &= \frac{(F_2 - F_3)(S - S_3) + (S_3 - S_2)(F - F_3)}{(F_2 - F_3)(S_1 - S_3) + (S_3 - S_2)(F_1 - F_3)} \\
\lambda_2 &= \frac{(F_3 - F_1)(S - S_3) + (S_1 - S_3)(F - F_3)}{(F_2 - F_3)(S_1 - S_3) + (S_3 - S_2)(F_1 - F_3)} \\
\lambda_3 &= 1 - \lambda_1 - \lambda_2
\end{aligned} \tag{2.4.1}$$

where F_i and S_i are the ith model's force and speed, F is the current force value, and S is the current speed value. The interpolated LSFs are then converted back into AR coefficients. New coefficients and variance values are calculated at 240 Hz.

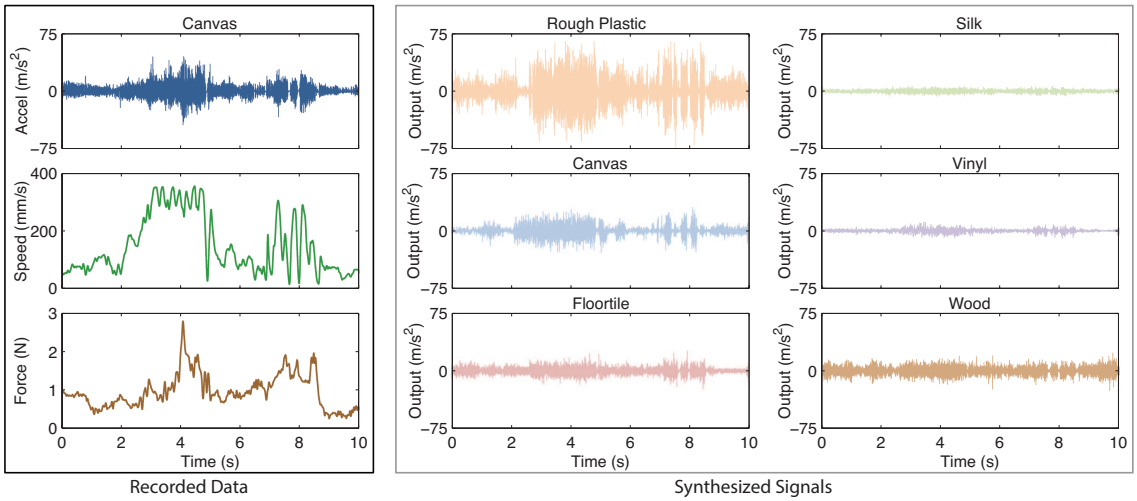

Figure 2.10: Recorded and synthesized data for comparison of texture model set performance. The acceleration, force, and speed data were recorded from the canvas material using the haptic recording device. The recorded force and speed signals were then used to synthesize accelerations for all six materials.

2.4.2 Signal Synthesis

A white Gaussian excitation signal is generated with power equal to the interpolated variance value. Histories of the excitation signal and vibration outputs are used to calculate the new vibration output at a rate of 10 kHz.

Fig. 2.10 shows synthesized texture vibration signals for the six chosen materials. Force and speed signals recorded while touching real canvas were used to synthesize acceleration signals for all six textures to show their varied output responses. The signal synthesized using the canvas model set has the best match to the recorded acceleration signal. However, the synthesized canvas signal is lower in power than the recorded canvas signal in places because the test trajectory included forces and speeds higher than those that were modeled.

2.5 Method Evaluation

After devising a way to model haptic textures from unconstrained tool-surface interactions, we sought to evaluate the quality of the resulting virtual texture models. The ideal output of any data-driven texture modeling approach is a virtual texture that is perceptually indistinguishable from the real texture on which it is based. The experience of a user interacting with such a texture depends on all of the components of the system, including the hardware on which the texture is rendered. To better assess the contributions of this Chapter, we focused our evaluation on texture signal generation, leaving human perceptual assessments of the interactively rendered textures for Chapter 3. After developing a method for comparing two virtual textures or a real texture and a virtual texture, we evaluated our virtual textures across six materials, across nine haptographers, and across fifty-four combinations of material and haptographer.

2.5.1 Spectral Difference Metric

Because each recording involves a unique force-speed trajectory, each virtual texture model set we create includes AR models at a novel set of forces and speeds. Thus it is not possible to directly compare two virtual texture models to one another, as done in [24]. Instead, we do such comparisons by choosing a force-speed trajectory, applying it to the two virtual textures, and comparing the resulting synthesized vibration signals using the metric described below. The same approach can also be used to

compare a certain recording of a real texture to what would be felt touching any virtual texture with the force-speed trajectory used during the recording.

When the selected force and speed vary over time, the texture acceleration signal is not stationary, so we compare the frequency spectra of the two signals for short windows of time and then average the results across all windows. Each ten-second signal was broken down into windows of 256 datapoints (25.6 ms) with 50% overlap, and a Hamming window was applied to prevent spectral leakage. The amplitude of the window's discrete Fourier transform (DFT) was then normalized by the total power in the window, so that the spectral shape of two signals could be compared, irrespective of signal power. The normalized spectral error (NSE) was used as a metric to compare the spectral shape of the two signals.

$$\mathrm{NSE}_j = \frac{1}{N_f} \sum_k |Y_j(f_k) - X_j(f_k)| \tag{2.5.1}$$

where N_f is the number of frequencies, and $Y_j(f_k)$ and $X_j(f_k)$ are the normalized amplitudes of the DFTs of the two signals being compared in the jth window at frequency f_k.

The NSE disregards differences in the power of the two signals, but power is important in determining the perceptual roughness of the surface. Therefore, we also compare signal power.

$$\mathrm{PE}_j = \frac{|P_{Y,j} - P_{X,j}|}{\frac{1}{2}(P_{Y,j} + P_{X,j})} \tag{2.5.2}$$

Here $P_{Y,j}$ and $P_{X,j}$ are the powers of the two signals being compared in the jth

window. Both the frequencies and the magnitude of a vibration are important in determining the feel of a surface. Therefore, we created the spectral difference (SD) metric to place equal emphasis on spectral shape and power.

$$\mathrm{SD} = \frac{1}{N_w} \sum_j \left(\frac{\mathrm{NSE}_j}{\mathrm{std(NSE)}} + \frac{\mathrm{PE}_j}{\mathrm{std(PE)}} \right) \tag{2.5.3}$$

where N_w is the number of windows, std(NSE) is the standard deviation of the normalized spectral error across all tested windows, and std(PE) is the standard deviation of the power error across all tested windows. Normalizing the NSE and PE yields a dimensionless metric that is scale invariant and weights the two factors equally.

2.5.2 Spectral Differences Across Materials

To establish an acceptable range for the SD metric, the first test evaluated the differences between virtual textures made for different materials. This test used virtual textures made from ten seconds of data recorded by the experimenter for each of the six real materials shown in Fig. 2.3. For comparison, the experimenter also made three separate recordings for each of the real surfaces, ensuring the forces and speeds spanned those commonly used when exploring textured surfaces with a tool. The

Table 2.1: Spectral Differences Across Materials

Model	T1	T2	T3	T4	T5	T6
VT1	4.50	5.33	6.29	7.11	7.48	6.41
VT2	5.46	3.81	4.54	5.79	5.52	5.12
VT3	6.04	4.81	3.91	4.92	4.68	4.28
VT4	6.62	6.21	5.11	4.44	4.47	4.96
VT5	7.13	5.64	5.13	4.53	4.04	5.57
VT6	6.05	5.27	4.27	5.05	4.95	4.23

T1: Rough plastic, T2: Canvas, T3: Floor tile, T4: Silk, T5: Vinyl, T6: Wood

maximum speed was 170 mm/s, and the maximum force was 2.8 N. The forces and speeds of each recording were used to drive all six of the experimenter's virtual textures, creating synthetic vibrations that could be compared across materials. The spectral difference metric was calculated for each pair of real vibrations and vibrations synthesized using that recording's force-speed trajectory. These differences were then averaged across all three trajectories to yield the results shown in Table 2.1. As anticipated, the shaded boxes along the diagonal reveal the relatively small spectral difference between the vibrations recorded from a real material and the vibrations synthesized from a virtual version of the same material. Two standard deviations above the mean of the differences along the diagonal is 4.72, which we will use as the baseline of the spectral difference metric. Furthermore, the spectral difference is also relatively small for material pairs of similar roughness and surface feature size, for example rough plastic and canvas or silk and vinyl.

2.5.3 Spectral Differences Across Subjects

The second test compares real vibrations recorded by one person to virtual textures made from data recorded by several different individuals touching the same material. We collected data from eight individuals dragging the recording tool across each of the six real textures. The procedures were approved by the Penn IRB under protocol 816548, and subjects gave informed consent. The subjects were all right handed with limited experience with haptic devices. They were instructed to use natural and fluid

motions and to vary force and speed as desired. Ten seconds of force, speed, and acceleration data were recorded for each subject and material, resulting in 48 recordings. Each of these recordings was segmented and modeled as described in Section 2.3. For comparison, we also consider the previous section's virtual textures, which were created from data recorded by the experimenter, an experienced haptographer.

The three force-speed trajectories for each material from the previous comparison were used to synthesize vibrations for this comparison. The spectral difference was calculated for each pair of recorded and synthesized accelerations, as shown in Table 2.2. As expected, the spectral differences for the experimenter have the smallest mean and standard deviation. Thus, the models made by the experimenter are more consistent when compared with data also recorded by the experimenter than models made by other subjects. This trend is due to differences in hand dynamics and grip, which affect the vibrational characteristics of the pen. Many of the SD values for the eight other subjects were within the baseline established above, as shown by the shaded boxes. High SD values are caused by individual differences in hand dynamics or because a subject's motion did not cover the entire tested region of force-speed space.

2.5.4 Spectral Differences Across Materials and Subjects

The final comparison evaluates the validity and consistency of the texture models created across all materials and haptographers. A single force-speed trajectory was

Table 2.2: Spectral Differences Across Subjects

Sub.	T1	T2	T3	T4	T5	T6	*mean*	*stdev*
S1	4.82	4.23	4.07	4.91	4.23	5.19	4.57	0.46
S2	5.53	4.64	4.04	4.54	4.68	4.63	4.68	0.48
S3	5.02	4.36	5.04	5.27	4.51	5.52	4.95	0.44
S4	5.42	4.19	3.94	4.92	4.04	5.28	4.63	0.66
S5	4.76	4.14	4.77	5.01	4.17	4.70	4.59	0.35
S6	4.44	3.76	3.96	5.26	4.39	4.99	4.47	0.58
S7	4.75	4.00	3.85	4.98	3.94	5.01	4.42	0.55
S8	4.61	3.42	3.81	4.26	4.54	4.31	4.16	0.46
Exp	4.50	3.81	3.91	4.44	4.04	4.23	4.15	0.28

chosen for this comparison. This trajectory was created by simulating a person's trajectory through the force-speed space using a smoothed random walk algorithm. This method was chosen over using a recorded force-speed trajectory to ensure that the trajectory fully spanned the force-speed space of interest. The trajectory was used to simulate 30 seconds of vibration using each virtual texture from the eight subjects plus the experimenter, which resulted in 54 synthesized signals. These signals were then compared pairwise (1431 pairs) using the spectral difference metric to create a dissimilarity matrix.

Multi-dimensional scaling (MDS), which is a common tool in haptic texture research for visualizing differences between materials [12, 119], was used to analyze the dissimilarity matrix. This technique optimally places the virtual textures in an N-dimensional perceptual space such that the distances between the points approximately follow a monotonic transformation from the dissimilarities. Analysis of the scree plot for up to six dimensions indicated that two or three dimensions were necessary to achieve a good fit. Fig. 2.11 shows the 54 synthesized signals in a two-dimensional perceptual space. The signals have been divided into six clusters using

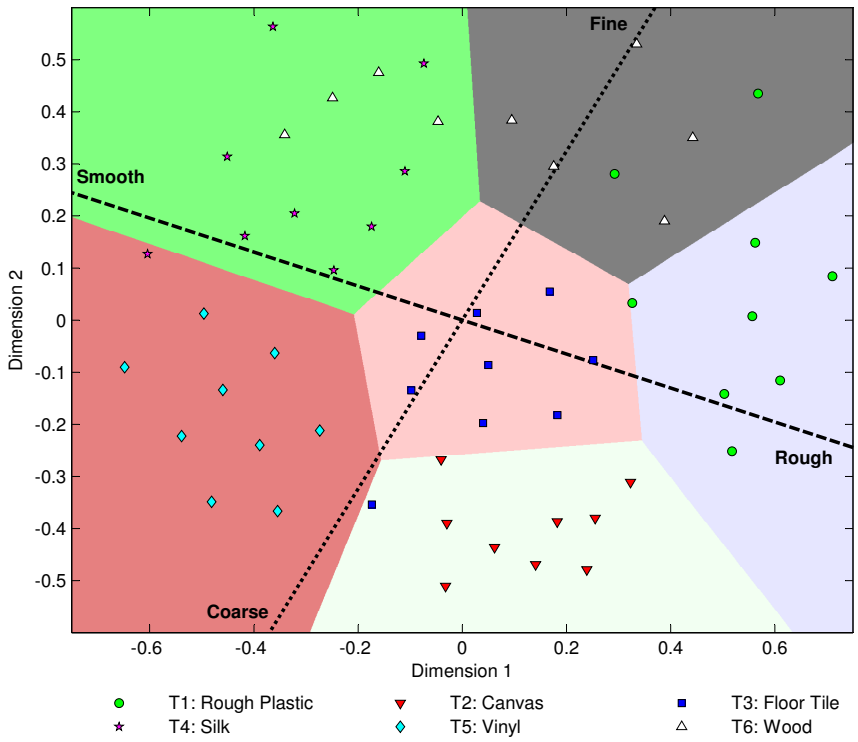

Figure 2.11: Texture models for all subjects clustered quantitatively using the spectral difference metric. Models were largely clustered according to material. Lines showing the perceptual qualities of roughness and fineness of surface features are displayed.

k-means clustering. The regions in the figure represent the decision boundaries of the clusters. As expected, synthesized signals from models of the same material are closely clustered. Some outliers exist due to differences in hand dynamics between subjects or if a subject's movement did not span a large portion of the force-speed space. Wood is not consistently placed in this two-dimensional space due to its non-isotropic behavior. If the tool is dragged perpendicular to the grain, it feels relatively rough. If the tool is dragged parallel to the grain, the result is a smoother texture. Since the subjects were not constrained to drag the tool in only one direction, the signals used to create models for wood contained vibrations from both the rough

and smooth textures of wood. These inconsistent vibrations created models for wood that are sometimes rough and sometimes smooth; future work should include tool direction for modeling non-isotropic materials.

The average windowed power and the average windowed spectral centroid (SC) were calculated for all 54 synthesized signals. The roughness of a material is captured most by the power of the vibration, as shown in [45,71,89,119]. The SC was proposed as a measure of fineness of surface features in [33] and is calculated as:

$$\mathrm{SC}_j = \frac{\sum_k X_j(f_k)^2 f_k}{\sum_k X_j(f_k)^2} \tag{2.5.4}$$

where $X_j(f_k)$ is the normalized amplitude of the DFT in the jth window at frequency f_k. The power and spectral centroid were averaged across all windows. Linear regression was conducted on the power and SC of the 54 signals to determine the relationship between the two perceptual qualities and the placement of the signals in the MDS space. The resulting perceptual quality lines are superimposed on the cluster diagram in Fig. 2.11. The lines for roughness and fineness are close to orthogonal with an angle of 76.4°, which indicates little correlation between the two qualities. They both contribute independently to the tactual perception of the materials.

The first MDS dimension closely follows the roughness of the materials. Rougher materials are located on the right side of the cluster diagram, and the materials get progressively smoother as you move to the left. Wood is spread across this rough-smooth dimension because of its non-isotropic behavior. The materials also follow

the fineness perceptual quality line as expected. Physically, canvas has the largest surface features, and silk has the smallest surface features of the material set, as can be seen visually in Fig. 2.3.

2.6 Summary

This research developed methods for the creation of haptic texture models from data recorded during unconstrained tool-surface interactions. First we presented our data-recording hardware, which measures the speed, force, and high-frequency acceleration of the tool as a user drags it across a texture. Since the user's force and speed vary during data capture, the acceleration signal is not stationary. Therefore, we segment the signal to represent it as a piecewise autoregressive process. This procedure results in a texture model set that represents the feel of the texture at many input conditions. Each texture model set is stored in a Delaunay triangulation with each vertex representing a single model. Measurements of force and speed are used to synthesize texture vibration signals.

A spectral difference metric was presented to quantify the fit of the models and the consistency of models across many subjects. The fit of models is sensitive to differences in individual hand dynamics, but the models made from short data recordings result in a reasonable spectral fit. Synthesized vibrations can be clustered by material based only on the spectral difference metric, and the spectrum encodes the material's roughness and fineness of surface features.

Chapter 3

Rendering and Evaluation of Haptic Textures

Although much research has been done in determining the perceptual qualities of real textures [45, 89, 119], there has not been extensive work in evaluating how these perceptual qualities translate to virtual textures. Past studies that have analyzed virtual textures focused on realism as a whole [95, 97]. In contrast, this Chapter seeks to assess the perceptual qualities and similarities of real and virtual textures to evaluate the strengths and weaknesses of the modeling approach that was presented in Chapter 2. We hope this investigation helps researchers make informed decisions about adopting similar methods. This Chapter presents work published in the IEEE Transactions on Haptics [26].

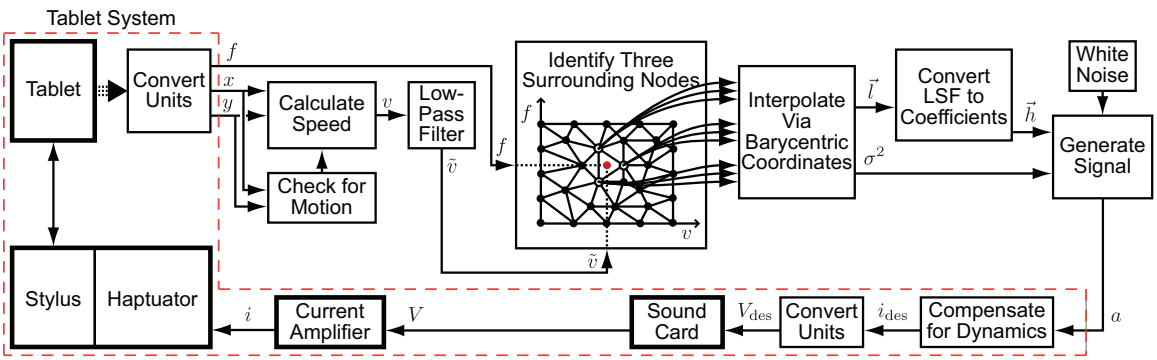

Figure 3.1: A diagram of our full texture rendering system. A texture vibration signal is synthesized using measurements of the user's current position and force given by the Wacom tablet. This signal is then output via the computer soundcard, through a current amplifier, to drive the Haptuator on the stylus. The dashed red box separates the tablet hardware and output system.

3.1 Rendering Vibration Signals

This Section describes the hardware and steps necessary to render synthetic textures to a human user utilizing the texture model set created in Section 2.3 and the texture vibration signal synthesized as in Section 2.4. Fig. 3.1 shows the full diagram of the texture synthesis and rendering process.

3.1.1 Tablet System

The virtual haptic textures are displayed on an updated version of the TexturePad described in [95] and shown in Fig. 3.2. This system consists of a Cintiq 12WX interactive pen display by Wacom Co., Ltd., which is capable of providing measurements of the user's force and position at a rate of 125 Hz. The Linux Wacom Project code base [2] provides access to these variables. The force sensor has a resolution of 0.0013 N, and the x-position and y-position measurements have a resolution of

Figure 3.2: Our texture rendering system. A force sensor in the stylus measures the applied normal force, and sensors embedded in the Wacom tablet measure the tool's position. The computer generates a synthetic texture vibration in real time, and a Haptuator mounted on the stylus transmits these vibrations to the user's hand.

4.934 μm and 4.861 μm, respectively. The position data is differentiated and low-pass filtered to obtain the user's speed. The force and speed measurements are used to generate synthetic texture vibration signals as described in Section 2.4.

3.1.2 Dynamic Compensation

After the next acceleration value is synthesized and before it is played to the user, a feed-forward dynamic compensation controller is implemented to diminish the resonance of the Haptuator and achieve a higher-fidelity recreation of the tool vibrations. This dynamic compensation scheme, which is outlined in [78], approximately inverts the system's dynamics to estimate the current output needed to create the desired tool vibration.

3.1.3 Rendering Hardware

The synthesized acceleration signal is output through the soundcard of the computer using the PortAudio C library [1]. This output voltage is then passed through a linear current amplifier with a gain of 1 A/V. The synthesized vibrations are transmitted to the user by driving a Haptuator vibrotactile transducer (TactileLabs, model no. TL002-14-A) that is firmly attached to the stylus via a custom 3D-printed plastic bracket, as shown in Fig. 3.2.

3.2 Texture Samples

The five surfaces used in the human-subject evaluation of our modeling and rendering methods are the same as those used in Chapter 2: rough plastic, canvas, floor tile, silk, and vinyl. Wood was excluded from the study due to its anisotropy described in Section 2.5.4.

Although they were not used directly in modeling, we determined the physical parameters of the materials so they could be compared to the rated parameters in the study reported later. The parameters of interest were roughness (surface feature height), hardness (stiffness), slipperiness (friction), and fineness (spatial frequency of surface features).

3.2.1 Roughness Measurements

The roughness of each material was found optically using the images shown in Fig. 2.3. The analyzed images were square with 500 pixels on each edge, which correlated to a physical size of 63.5 mm. All pictures were taken from overhead using a Nikon D40 digital camera mounted on a tripod under fluorescent lighting. The images were converted to grayscale and separated into their roughness and waviness components using a 2D Gaussian low-pass filter ($\sigma = 2.54$ mm). In the roughness component, a light pixel represents a high point (a peak) and a dark pixel represents a low point (a valley) on the texture. The presence and height of surface features (bumps) on the surface can be determined by comparing the grayscale values of adjacent pixels [35]. To estimate the roughness of the textures, which is a measure of the surface feature height, we calculated the symmetric grey-level co-occurrence matrix (GLCM) for each image. The GLCM is a measure of how often combinations of pixel grayscale values occur in an image. We divided the range of grayscale values into 500 equally spaced bins and calculated the GLCM for four directions $(0°, 45°, 90°, 135°)$, which were then summed to obtain a spatially invariant GLCM for the image. The GLCM was normalized by the sum of all components to express it as a probability. Using the normalized GLCM, the roughness is estimated as the Angular Second Moment (ASM):

$$\mathrm{ASM} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} M_c\left(i,j\right)^2 \tag{3.2.1}$$

Table 3.1: Physical Measurements of Materials

Material	Roughness (unitless)	Stiffness (R^2) (N/mm)	Friction (R^2)	Fineness (mm^{-1})
T1: Rough Plastic	0.3252	42.20 *(0.99)*	0.18 *(0.74)*	1.25
T2: Canvas	0.3515	17.82 *(0.98)*	0.36 *(0.87)*	1.04
T3: Floor tile	0.3702	10.81 *(0.99)*	0.32 *(0.85)*	0.98
T4: Silk	0.2776	37.86 *(0.98)*	0.30 *(0.86)*	1.46
T5: Vinly	0.2842	15.07 *(0.99)*	0.49 *(0.96)*	1.18

where i is the row number, j is column number, and M_c is the normalized GLCM [35]. The estimated roughness parameters, R_a, are shown in Table 3.1. Although these values are not direct physical measurements of the surface, they provide an accurate representation of surface roughness and allow for the roughnesses of different materials to be directly compared.

3.2.2 Surface Stiffness Measurements

Surface stiffness was found by measuring the force-displacement curve of each material. An MTS LSB.501 50 N load cell was attached to an MTS Criterion Model 43 load frame with a position resolution of 0.06 μm. A stainless steel 1.6 mm hemispherical tooltip was attached to the load cell, which was then lowered onto the material until a load of 10 N was achieved. Position and force measurements were recorded at 10 Hz. Due to mechanical compliance in the system, only data above 1.0 N was used in the analysis. Similarly, data beyond 3.0 N was discarded because the acrylic backing strongly biased the stiffness measurements for large forces. Furthermore, 3.0 N was the maximum force typically applied by humans when interacting with a texture

through a tool [25]. This range of forces is similar to that used to calculate surface compressibility in [12]. A line was fit to the force-position data between these two limits. The slope of the line is a measurement of the surface stiffness in N/mm, as shown in Table 3.1. No nonlinear behavior was observed, as indicated by the high R^2 values.

3.2.3 Friction measurements

The slipperiness of the materials, which can be represented as the coefficient of kinetic friction, was found using data recorded by the haptic recording device. The experimenter dragged the tool across the surface in a fluid motion and recorded ten seconds of interaction data. The force measurements were separated into the tangential and normal components, and a line was fit to the plot using a robust fit method in which outliers were removed. The slope of the line is the kinetic friction coefficient, as shown in Table 3.1. The chosen materials closely follow the proposed Coulomb friction model for the forces used, as indicated by the R^2 values.

3.2.4 Fineness Measurements

The images from above were also used to calculate fineness, the materials' spatial periodicity, as the average spatial frequency of the surface features. Each row and column of pixels was transformed into the frequency domain by taking the discrete Fourier transform (DFT) of the grayscale pixel values. This procedure gave frequen-

cies in units of pixel^{-1}, which were converted to absolute spatial frequencies in mm^{-1}. The DFT was averaged across all rows and columns, and the spectral centroid was calculated. The spectral centroid has commonly been used as a measurement of the fineness of a texture [25, 33]. The fineness parameters are shown in Table 3.1.

3.3 Human Subject Study

This section describes the human-subject experiment we ran to evaluate the realism of our virtual textures and the strengths and weaknesses of this approach. In particular, we wanted to determine the extent to which our modeling and rendering approach captures the roughness, hardness, slipperiness, and fineness perceptual dimensions. To accomplish this, we adapted the study and analysis conducted in [119]. All procedures were approved by the Penn IRB under protocol 816548, and subjects gave informed consent. Two separate sets of ten subjects were recruited for a total of twenty subjects. The subjects were all right-handed with limited experience with haptic devices.

3.3.1 Experimental Set-up

The subject sat at a table in front of the tablet and across from the experimenter. As shown in Fig. 3.3, a black curtain was hung between the subject and the tablet, and the subject wore headphones playing pink noise. These interventions ensured that the subject used only haptic, and not visual and auditory, cues during the study. The

subject used the augmented stylus to interact with the real and virtual textures. The tablet measured the user's force and speed during all texture interactions. When a virtual texture was present, the force and speed values were used to render the texture vibration signal, as described in Section 3.1. When a real texture was present, the Haptuator was not activated. An three-axis ADXL345 digital accelerometer configured into ± 78.4 m/s^2 (± 8 g) mode with a resolution of ± 0.153 m/s^2 was rigidly attached to the stylus and polled at 800 Hz to record the tool vibrations. Subjects were instructed to keep the stylus on the surface at all times except when switching between textures and were not allowed to tap or statically press the stylus into the surface. The experimenter monitored the subject's hand to ensure that it did not come in contact with the texture or tablet. A 0.6 cm thick acrylic piece with two 10.2 cm square cutouts was mounted to the face of the tablet. The cutouts were used to secure the materials during the study and to act as a guide to keep the stylus constrained to the textures. For virtual textures, a 0.32 cm thick smooth piece of acrylic was placed in the cutout on the tablet screen so thickness cues could not be used to determine the presence of a virtual texture.

3.3.2 Experimental Procedure

There were three separate phases to the study: free exploration, pairwise comparisons, and adjective ratings. One set of ten subjects completed the pairwise comparison and adjective ratings phases of the study, but that adjective rating data was not used in

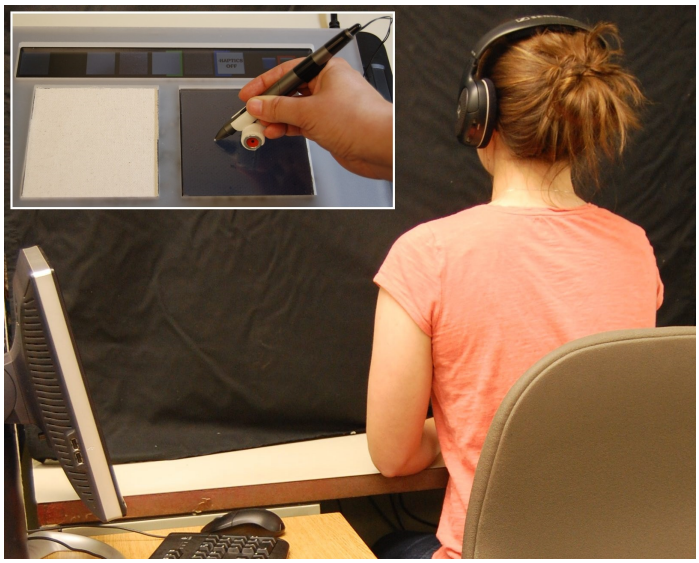

Figure 3.3: Experimental set-up. The subject sat in front of the tablet, across from the experimenter. A black curtain prevented the subject from seeing the textures, and headphones playing pink noise masked audio cues. The subject grasped the stylus augmented with a Haptuator in her dominant hand and interacted with real and virtual textures placed on the tablet screen.

this analysis because order effects may have been present. A second set of ten subjects completed the free exploration and adjective rating phases. The study used the real and virtual versions of five of the textures shown in Fig. 2.3, for a total of ten textures (five real, five virtual). Wood was used only in the pairwise comparison phase of the study, but the data associated with the real and virtual versions of wood were not used in the analysis due to its anisotropy, which was discussed in Chapter 2. Study data were collected and managed using Research Electronic Data Capture (REDCap) tools hosted at the University of Pennsylvania [41].

Free Exploration

In this phase of the study, the ten textures were presented to the subject one at a time in randomized order. At the beginning of each trial, the experimenter placed one texture in the right cutout on the tablet and then turned on an LED to signal that the texture was ready. The subject placed his or her hand under the curtain and was given the stylus, which he or she used to explore each texture for as long as desired. The first ten seconds of acceleration, force, and speed interaction data, as measured by the tablet, were recorded for each texture. The subject then removed his or her hand from under the curtain and typed a short description of that texture on a nearby computer. This procedure was repeated until the subject had felt all ten textures. This phase allowed the subjects to become familiar with the textures and the experimental set-up.

Pairwise Comparisons

In this study phase, all possible pairs of textures (real-real, virtual-virtual, and real-virtual) were presented to the subject one at a time for a total of 45 pairs. The textures were presented in randomized order with equal likelihood of being placed in the left or right cutout. For each pair, the experimenter placed the two textures in the cutouts on the screen and turned on an LED to signal that the textures were ready. The subject felt both textures with the stylus, switching between the two as much as desired. Subjects then rated the similarity of the two textures by placing a

mark on a scale of "completely the same" to "completely different" on the computer using a mouse wielded in their non-dominant hand.

Adjective Ratings

In the final phase of the study, the textures were presented to the subject one at a time in randomized order. In addition to the ten textures described above, the subjects were also presented with an acrylic piece without a virtual texture present. The subject was asked to rate each texture on the scales of rough-smooth, hard-soft, slippery-not slippery, and fine-coarse. These scales were chosen from the list provided in [81] as the psychophysical dimensions applicable to the perception of textures through a tool. The subject followed the same procedure as above but felt only the texture in the right cutout. The experimenter did not define the adjectives, so the subject had to rely on his or her intuitive understanding of the scales. The adjective scales were presented in randomized order, and the subject rated all textures along each scale before proceeding to the next scale.

3.3.3 Results

Free Exploration Recordings

The acceleration recordings from the free exploration phase of the study were used to compare the power and spectral centroid of the accelerations experienced when interacting with real and virtual versions of the same texture. The force and speed

recordings were also compared to determine if there was any difference in the speeds and forces commonly used by subjects touching real and virtual textures. The mean and one standard deviation above and below the mean of the acceleration power, acceleration spectral centroid, speed, and force for each ten-second-long recording were found. These values were combined across all subjects for each texture. The resulting two-dimensional box plots are shown in Fig. 3.4, allowing for direct visual comparison of the recorded values for corresponding real and virtual textures. The central lines of each box denote the real and virtual medians, the edges of each box denote the 25th and 75th percentiles, and the whiskers extend to the extreme data-points not considered outliers. A line denoting the ideal relationship between the real and virtual ratings is shown. Two-way analysis of variance (ANOVA) was performed separately on the power, spectral centroid, speed, and force values using texture and virtual/real as factors. The differences in the power of the acceleration signals were found to be statistically significant across different textures ($F = 101.2$, $p < 0.001$), but not statistically significant when comparing real and virtual textures ($F = 2.48$, $p = 0.12$). Similarly, the differences in the spectral centroid of the acceleration signals were found to be statistically significant across different textures ($F = 72.52$, $p < 0.001$), but not statistically significant when comparing real and virtual textures ($F = 2.31$, $p = 0.13$). No statistically significant differences were found in the speeds used by the subjects across different textures ($F = 0.48$, $p = 0.75$) and comparing real and virtual textures ($F = 1.48$, $p = 0.22$). Similarly no statistically significant

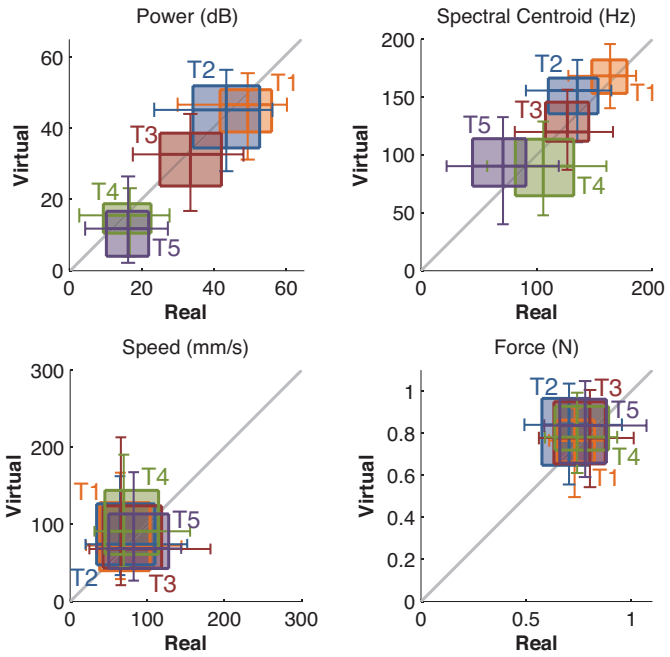

Figure 3.4: Comparison of the power and spectral centroid of the recorded acceleration signal and the speeds and forces used by subjects during the free exploration phase of the study. There is a strong linear trend for power and spectral centroid comparing values for the corresponding real and virtual textures. There is no statistical difference in the speeds and forces used by subjects on real and virtual textures.

differences were found in the forces across different textures ($F = 1.19$, $p = 0.32$) and comparing real and virtual textures ($F = 2.33$, $p = 0.13$). The free response descriptions provided by the subjects during this phase were not analyzed because their purpose was to allow subjects to form an unbiased perception of the textures before being asked to rate them along the four adjective scales.

Pairwise Comparisons

Each subject's pairwise dissimilarity matrix was normalized by the standard deviation of their dissimilarity ratings to compensate for individual differences in subjective

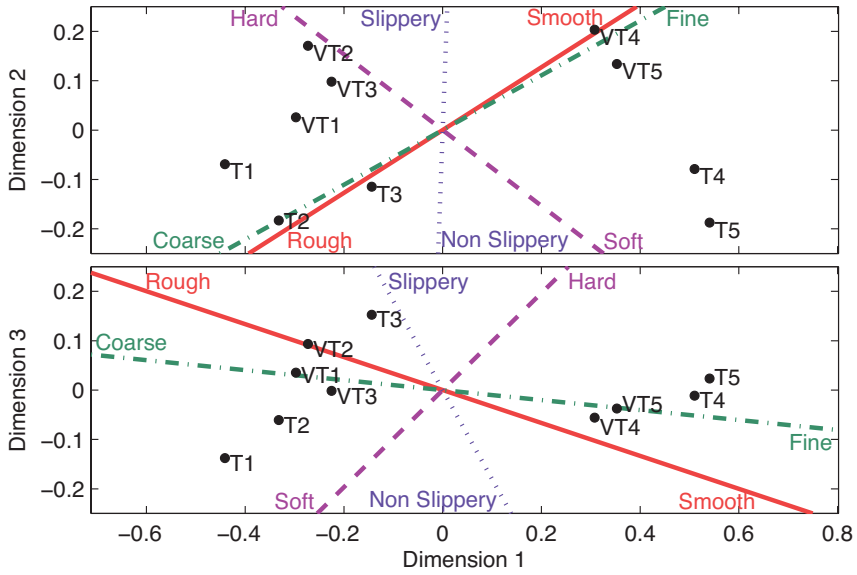

Figure 3.5: Real and virtual textures placed in three-dimensional perceptual space based on dissimilarity ratings. Lines showing the perceptual qualities of roughness, hardness, slipperiness, and fineness were fit to the space using ratings along the four adjective scales.

rating scales. The normalized dissimilarity matrices were averaged across the ten subjects. As in Section 2.5.4, multi-dimensional scaling was used on the resulting average dissimilarity matrix to place the ten textures in an N-dimensional perceptual space. The distance between a pair of textures in the perceptual space is proportional to their average rated dissimilarity. An analysis of the scree plot for up to six dimensions indicated that two or three dimensions were necessary to achieve a good fit ($R^2 > 0.9$). Fig. 3.5 shows the resulting fit of the ten textures into a three-dimensional perceptual space.

Fig. 3.6 shows the normalized dissimilarity ratings before averaging, separated into the four types of texture pairs (real-virtual same texture, real-real different tex-

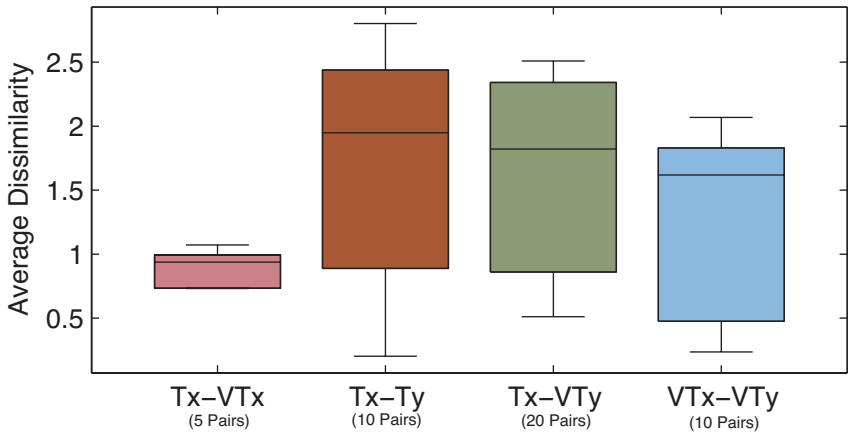

Figure 3.6: Comparison of dissimilarity ratings across the four types of texture pairs. The median dissimilarity rating for the real-virtual same texture pairs is significantly lower than the median ratings for the three other types of pairs.

tures, real-virtual different textures, virtual-virtual different textures). The median dissimilarity rating for the real-virtual same texture pairs is much lower than the median ratings for the three other types of pairs, indicating that the corresponding real and virtual texture pairs were rated as more similar than pairs of different textures.

To understand how the textures were distributed in the perceptual space, a cluster analysis was conducted on the Euclidean distances between textures. A hierarchical binary cluster tree was generated, and the resulting dendrogram is shown in Fig. 3.7.

Adjective Ratings

The adjective ratings were separately normalized for each subject along the four scales by dividing by the standard deviation along that scale for that subject. The normalized adjective ratings were then averaged across all subjects. The normalized adjective ratings before averaging are shown in the two-dimensional box plots in

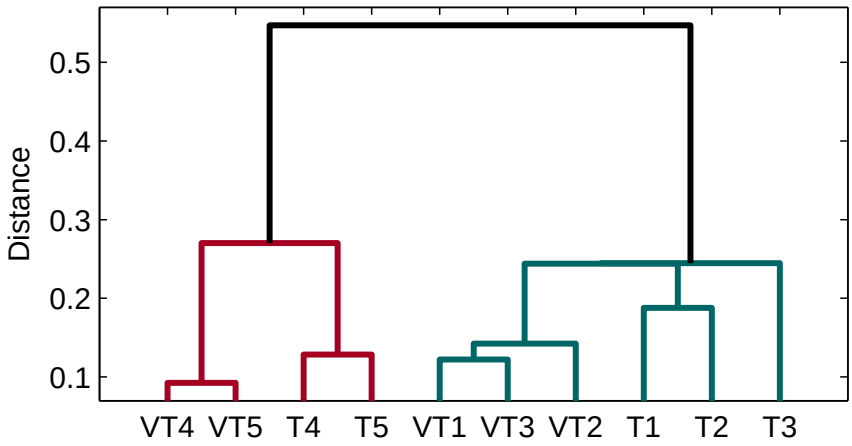

Figure 3.7: Dendrogram of materials showing clustering of real and virtual textures using the distances between textures in three-dimensional perceptual space.

Fig. 3.8. A two-factor ANOVA was performed separately on the four sets of adjective ratings using texture and virtual/real as factors. When comparing across textures, the differences in ratings of roughness ($F = 52.18$, $p < 0.001$), hardness ($F = 24.46$, $p < 0.001$), and fineness ($F = 44.76$, $p < 0.001$) were found to be statistically significant. The differences in slipperiness ratings were not found to be statistically significant across textures ($F = 1.74$, $p = 0.15$). When comparing virtual and real textures, the differences in hardness ($F = 9.72$, $p < 0.005$) and slipperiness ($F = 99.91$, $p < 0.001$) ratings were found to be statistically significant. The differences in roughness ($F = 2.42$, $p = 0.12$) and fineness ($F = 2.37$, $p = 0.13$) ratings were not found to be statistically significant when comparing virtual and real textures.

We separately regressed the four sets of average adjective ratings onto the perceptual space to place the four adjective lines, as shown in Fig. 3.5. In the regression, each set of adjective ratings was used as the dependent variable with the coordinates

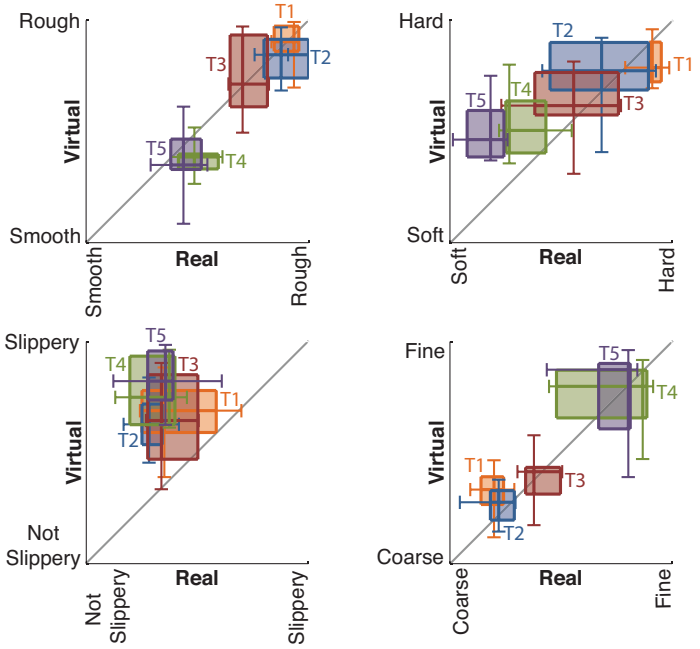

Figure 3.8: Comparison on subjects' ratings of the ten textures along the four adjective scales. The box plots for roughness and fineness show a strong linear trend in the ratings for the corresponding real and virtual textures. The box plot for hardness shows a similar trend in the ratings, while the box plot for slipperines shows no correlation.

Table 3.2: Correlation of Adjective Lines

Adjective	Roughness	Hardness	Slipperiness	Fineness
Roughness	1.00	0.81	−0.25	−0.98
Hardness	0.81	1.00	0.22	−0.83
Slipperiness	−0.25	0.22	1.00	0.24
Fineness	−0.98	−0.83	0.24	1.00

of the textures in the perceptual space as predictor variables. The independence of the adjective lines was examined by analyzing the correlations in Table 3.2. There was a high degree of negative correlation between the roughness and fineness scales, and between fineness and hardness ($p < 0.005$). There was also a high degree of positive correlation between roughness and hardness ($p < 0.005$). No other correlations were statistically significant.

To determine the correlation between perceived dissimilarity of the textures and the ratings along the four adjective scales, we performed a multilinear regression on the average dissimilarity ratings. The pairwise differences in the average ratings of textures along the four adjective scales were used to predict dissimilarity, $\hat{D}$:

$$\hat{D} = b_1 d_r + b_2 d_h + b_3 d_s + b_3 d_f \tag{3.3.1}$$

where $b_1 = -0.081$, $b_2 = 0.178$, $b_3 = 0.071$, and $b_4 = 0.366$ are the regression coefficients, and d_r, d_h, d_s, and d_f are the respective differences in the roughness, hardness, slipperiness, and fineness between textures. Fig. 3.9 shows the normalized perceived versus predicted dissimilarity. There was found to be a high degree of correlation ($R^2 = 0.88$) between the predicted and perceived dissimilarity values. Therefore, approximately 88% of the dissimilarity between the textures can be explained by the differences in the perceived roughness, hardness, slipperiness, and fineness of the textures.

Comparing the adjective ratings to the measured parameters in Table 3.1, the subjects performed best at discriminating the differences in roughness of the real textures ($R^2 = 0.54$) and marginally worse at discriminating differences in slipperiness ($R^2 = 0.31$) and fineness ($R^2 = 0.29$) although none of these correlations are statistically significant. The rated hardness of the textures shows the greatest difference from the relative measured hardness ($R^2 = 0.13$).

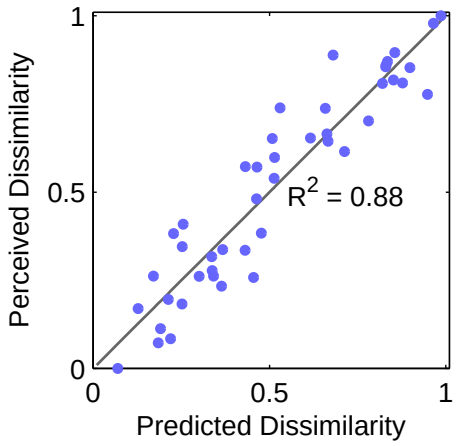

Figure 3.9: Normalized perceived vs. predicted dissimilarity. The perceived dissimilarity is found using the dissimilarity ratings averaged across all subjects. The predicted dissimilarity is found by performing multilinear regression on the average dissimilarity ratings using as predictors the pair-wise differences in the average ratings of textures along the four adjective scales.

3.3.4 Discussion

By evaluating how humans perceive both real and virtual textures, the study provided many important insights into the strengths and weaknesses of our modeling and rendering techniques. Since it was shown that the speeds and forces used by subjects were not statistically different across textures, the perceived dissimilarities and the corresponding adjective ratings were most likely not influenced by different interaction conditions. The power and spectral centroid of the resulting vibrations, however, were statistically different across textures and were presumably used by subjects during rating.

The dendrogram in Fig. 3.7 shows that there are two distinct clusters separating virtual and real silk and vinyl from virtual and real rough plastic, canvas, and floor tile. Comparing this clustering to the adjective ratings in Fig. 3.8 and the measured

physical parameters in Table 3.1, one can conclude that the main distinction between the two groups is their respective roughnesses. Furthermore, from the ANOVA analysis on the adjective ratings along the four scales, we determined that the distinction between virtual and real textures is not statistically significant when comparing the rated roughness of the surfaces. Thus, our modeling and rendering methods accurately capture and recreate the perceptual feeling of roughness of real textures, which supports the use of tool vibrations to represent roughness [10, 11, 58, 119]. However, textures that were rated rough were also commonly rated coarse, and textures rated smooth were also rated fine. Although roughness and fineness were used as separate perceptual dimensions in previous research [33], the high level of correlation between the two reveals that subjects were not able to effectively separate the dimensions in this study. This interaction is supported by the study ran in [13], which indicated that both physical roughness and fineness contribute to a subject's personal definition of perceived roughness.

Although the clustering of the textures along the rough/smooth parameter shows promise for our methods, the real and virtual versions of the same textures are not directly clustered together. This disparity indicates that other perceptual parameters that are not being directly modeled and rendered are important in the realism of virtual textures. Differences in the rated hardness of textures were significant between real and virtual surfaces, but Fig. 3.8 does show similar trends across the five textures of both types. Since all of the virtual textures were rendered on the same piece of

acrylic, and since hardness was not directly modeled or rendered, one might expect all of the virtual textures to be rated the same hardness as the acrylic piece, which was hardest of the materials (52.1 N/mm). Since the virtual textures received different hardness ratings, the vibrations that were played to the subject seem to have created an illusion of differing hardnesses, which we would expect because of the interplay between the vibrational characteristics of the tool-hand-surface system and the surface stiffness. More tests would be needed to confirm the effect of data-driven texture vibrations on the perceived hardness of a surface. However, to fully capture the hardness dimension, the surface stiffness must be rendered separately using a force-feedback device. Additionally, the hardness ratings of the real textures showed the lowest correlation to the measured stiffness parameters among the four parameters. This inconsistency may be due to the fact that subjects were not allowed to tap or statically press into the surface, which are the motions humans typically use to evaluate the hardness of an object [65].

Further variation in the clustering can be explained by the slipperiness ratings. Each virtual texture was consistently rated as more slippery than its corresponding real texture, as indicated in Fig. 3.8. However, there was not significant variation in the rated slipperiness when comparing across textures for either real or virtual textures. This similarity is not supported by the measured coefficients of friction in Table 3.1, indicating that there were factors other than friction affecting the perception of slipperiness of the real textures. When a tool is dragged across a real rough

surface, the tooltip catches on the bumps, which cause impulses to the tooltip in addition to the friction of the material. Thus, it may have been difficult for subjects to separate the slipperiness of a real texture from its roughness [11], and the higher slipperiness ratings of the virtual textures may be due partly to their inability to provide these hard stops. This discrepancy in slipperiness may also be partly due to the choice of the acrylic piece as the base on which to render the virtual textures. The acrylic piece was rated as the most slippery, which agrees with its measured coefficient of friction ($\mu_k = 0.13$). Choosing a less slippery base material would potentially increase the slipperiness match between real and virtual by decreasing the slipperiness of the virtual textures, but fully capturing slipperiness requires that friction be rendered directly, which cannot be done with the current tablet system.

3.4 Summary

This Chapter presented a method for using texture model sets to create realistic virtual textures in real time. The TexturePad system measures a user's force and speed and outputs a spectrally accurate time-domain vibration signal to a Haptuator.

We ran a study that evaluated the realism of our virtual textures and the strengths and weaknesses of our modeling approach. Subjects rated the similarity of pairs of textures. Subjects also rated all textures along four adjective scales (roughness, hardness, slipperiness, fineness). The study indicated that our virtual textures accurately capture and recreate the roughness of the real textures. However, the rated hardness

and slipperiness of the virtual textures did not correlate well with the real textures. Therefore, hardness and slipperiness should be added to the rendering process to improve the realism of our virtual textures. Our current TexturePad system is incapable of modulating stiffness or friction, so a different rendering platform must be implemented to provide these other aspects of the textures. Although our modeling method has been shown to work well for isotropic, homogeneous textures, this is still a limited subset of the types of textures that are commonly encountered. Thus additional modeling considerations are needed to expand the utility of our methods to model anisotropic and periodic textures as well.

The recording, modeling, and rendering methods presented in Chapters 2 and 3 were demonstrated at the 2013 IEEE World Haptics Conference in Daejeon, South Korea to approximately 300 attendees [23]. Individuals were encouraged to bring their own isotropic surfaces and make their own texture recordings, which we then used to create a virtual version of their texture. Approximately 50 attendees participated in both the recording and modeling part of the demonstration, and the remaining attendees tried only the rendered virtual textures. This demonstration won the Best Hands-On Demonstration award by audience vote (out of 80 demonstrations).

Chapter 4

Three-Dimensional Rendering of Haptic Surfaces

The field of haptic rendering seeks to understand the sensations felt during physical interactions to create realistic three-dimensional virtual objects that can be felt as well as seen. Traditional haptic rendering algorithms and devices, however, are incapable of outputting high-fidelity reproductions of surface contacts [17]. Physics-based simulations of textured surfaces are too computationally complex for real-time haptic rendering [85]. Therefore, surfaces are commonly represented using a Hooke's law relationship with output force proportional to the user's penetration into the object. The maximum surface stiffness an impedance-type haptic device can stably render is limited by the system's sampling rate, computational time delays, and quantization of position measurement and force output in the device [31]. These limitations cause

hard virtual objects to feel unrealistically soft and spongy. Friction and flexibility in the device limit its ability to accurately reproduce high-frequency vibrations [17], meaning that virtual surfaces often do not include texture and thus feel smooth and slippery.

To overcome the limitations of physics-based simulation, many researchers have sought to capture the complexities of physical interactions through data-driven modeling methods, which rely on data recorded during the real interaction of interest [83]. Data-driven modeling methods have shown promise in creating realistic virtual sensations, but most focus on modeling only one component of the surface, as summarized in Section 2.1. For example, Chapters 2 and 3 present our methods for creating virtual haptic textures by modeling and rendering the high-frequency vibrations that result from dragging a tool across textured surfaces at different forces and speeds. The human subject study in Section 3.3 showed that these vibrations accurately presented the roughness information of the surface, but they did not capture any of its slipperiness information and did not fully capture its hardness. This result indicated that other modeling and rendering components are necessary to fully capture and recreate the haptic sensations felt when touching surfaces with a tool. The tablet system we created for rendering textures cannot modulate surface friction or stiffness, so we needed to develop a different rendering platform. We began this process by extending our rendering approach to display textures and surface friction on a three-dimensional object using a Sensable Phantom Omni [21]. This rendering software was released in

our open-source Penn Haptic Texture Toolkit, which is discussed in Appendix A.

One benefit of using an Omni is that surface stiffness can be directly controlled. However, the Omni cannot output surface stiffnesses higher than 2.3 N/mm [3], which is much lower than the stiffness of many real objects. Furthermore, the maximum stable stiffness is actually less than the Omni's specified maximum stiffness [31]. Therefore, simply modulating the stiffness of the virtual surface is not sufficient for accurately portraying hardness. Displaying the high-frequency transient vibrations that occur when tapping on a surface has been shown to increase the perceived hardness of virtual objects and provide a better perceptual match with real surfaces than Hooke's law alone [62].

This Chapter builds upon our past work in texture rendering discussed in Chapters 2 and 3 to capture the additional surface properties of slipperiness and hardness. Our haptic surface models are comprised of three separate components: friction, tapping, and texture. The individual model components are based on acceleration, force, and motion data that are recorded when interacting with the physical surfaces through a tool. We render these realistic tool-mediated virtual haptic surfaces through a combination of force and vibrotactile feedback.

We evaluate these haptic surfaces through two separate human-subject studies. Section 4.5 discusses a study we designed and ran to evaluate the importance of creating texture vibrations that respond to user force and speed. This study investigates the individual effect of these two user inputs on the realism of our virtual textures.

Section 4.6 discusses a second study we designed and ran to evaluate the improvements in realism that can be achieved by adding friction, tapping, and texture cues to a virtual surface. This study assesses the contribution of the three components of the haptic surface model and determines whether all three model components are needed to create realistic virtual versions of each surface. This Chapter presents work published in the Proceedings of the IEEE Haptics Symposium [20] and work that was submitted for publication in the IEEE Transactions on Haptics [19].

4.1 Background

Traditional rendering schemes cannot fully capture the haptic sensations a person experiences when touching physical objects. These rendering algorithms present the shape of objects through a linear Hooke's law relationship that relates the output force to the user's penetration into the object [120], but they are missing other important haptic cues. When contacted through a tool, objects can be characterized by three main perceptual components: slipperiness, hardness, and roughness [81]. To capture these three perceptual dimensions, researchers have sought ways to accurately model and render surface friction, tapping behavior, and texture vibrations. Since haptic rendering schemes run in real time, they must model haptic interactions through a simplification of the underlying physics. To achieve this simplification and create virtual haptic interactions that feel realistic, many researchers have implemented data-driven methods, which seek to capture the output response of a system given

user inputs [83].

The perceptual dimension of slipperiness has been found to originate from surface friction [81]. Kinetic friction is modeled as the tangential force that resists motion when a tool is dragged across a surface with a certain normal force and tangential speed. In addition, some friction models include a region of static friction, which resists the start of motion. Several friction modeling techniques of differing complexity have been explored for use in haptic rendering. Richard et al. presented a method for modeling the friction between an aluminum block and three different surface materials from measured force, acceleration, and velocity data [93]. After considering several candidate models, Richard et al. chose a modified Karnopp friction model. Worden et al. also explored modeling pre-sliding and sliding friction by testing a number of grey-box and black-box models using friction data that was recorded using a tribometer [115]. Jeon et al. fit force, position, and velocity data to a Dahl friction model, which was then used to render friction in a haptic augmented reality scheme [51]. In previous work, we modeled the friction between a hemispherical tool and a textured surface using a modified Coulomb friction model that included a region of viscous friction around zero velocity [21]. Many different friction models have been proposed, and the behavior of the surface friction varies greatly depending on the properties of the two surfaces in sliding contact. Therefore, an appropriate friction model must be chosen by observing the behavior of the individual system. During rendering, friction is typically displayed to the user through a force-feedback haptic device.

Researchers have traditionally believed that perceptual hardness is related to the elasticity or compressibility of a surface [81]. However, those studies focused on the human perception of hardness using a bare finger. LaMotte studied the perception of hardness through a tool and found that humans were significantly better at discriminating the hardness of surfaces when tapping rather than when statically pressing the tool into the surface [65]. This result indicates that the transient vibrations elicited from tapping a tool on a surface largely determine the surface's perceived hardness. Many methods have been explored to model these transient signals to change the perceived hardness of virtual surfaces. These methods typically follow either an input-output or database approach [83]. In an input-output approach, a mathematical relationship is found between the system parameters (e.g., material type, incoming velocity) and the system's output (e.g., measured vibrations). For example, Wellman and Howe approximated the force response of a stylus tapped on a rigid surface as an exponentially decaying sinusoid with a frequency that depends on the material [112]. Okamura et al. extended this work by creating a library of tapping responses for multiple materials and simultaneously displaying normal force and tapping vibrations [82]. In a database approach, a set of transient vibration signals are recorded from repeated taps on the surface and are stored directly. Kuchenbecker et al. [62] ran a study to compare the realism of virtual surfaces using a database approach, an input-output approach, and Hooke's law. The study showed that directly playing either the velocity-scaled recorded acceleration transients or manually tuned

decaying sinusoids created a virtual surface that most closely matched the perceived hardness of the real surface. Displaying event-based tapping transients through a force-feedback device has been shown to significantly increase the perceived hardness of a virtual surface over the use of proportional feedback alone, but it's not yet known whether softer surfaces should have tapping transients.

The perceptual roughness of surfaces felt through a tool is known to depend on the vibrations induced during dragging [10, 13, 58, 119]. Following the principle of distal attribution, you recognize that these vibrations come from the contact between the tool and the surface [74]. During contacts with real textures, the amplitude and frequency of these vibrations change with normal force [70] and exploratory speed [71]. Logically, virtually rendered textures designed to simulate real surfaces should respond to the user's motions in the same manner. Loomis has asserted that to achieve full distal attribution, the signals that are felt must correspond to the user inputs in a physically appropriate manner [74]. Many researchers have explored this problem of creating realistic data-driven texture vibrations models as a function of user input.

Several of these texture rendering approaches have focused on modeling textures as a function of user speed. Vasudevan and Manivannan modeled the frequency spectrum of the vertical perturbations from dragging a Phantom Omni across a textured surface [107]. They recorded the spectrum for two tangential speeds and interpolated between them during rendering. Two groups have also directly played back recorded

texture vibrations using a cable-driven system [97] and a voice-coil actuator [102]. In these two examples, the recorded vibrations were sped up or slowed down depending on user speed.

More sophisticated texture modeling techniques have created haptic textures that depend on both user force and speed. Okamura et al. represented patterned textures as a set of data-driven decaying sinusoids at the material's characteristic frequency [82]. During rendering, the amplitude of the output vibrations depended on user force and speed for both stochastic and patterned textures; the frequencies depended on tool speed for only patterned textures. Guruswamy et al. created texture models based on a spatial distribution of infinite-impulse-response filters that are fit with decaying sinusoids [36]. The amplitude of the acceleration profile varied with normal force, and the rate at which the modeled impulse forces were played depended on user speed. In past work, we have also modeled texture vibrations as a function of user force and speed using time series models [21, 24–26, 95, 96]. We have postulated that the perceived realism of these virtual textures stems from the fact that they respond to both force and speed, but we had never tested this hypothesis. Furthermore, researchers have not previously studied whether all surface models need to include texture.

Past work in haptic modeling and rendering has tended to focus on capturing only one perceptual dimension. However, a surface's slipperiness, hardness, and roughness are all important in creating the full perceptual feel of the surface. This Chapter

seeks to create realistic haptic surfaces by combining friction, tapping, and texture.

4.2 Surface Modeling

The haptic surface models presented in this Chapter are composed of three separate modeling components: surface friction, tapping transients, and texture vibrations. These three components seek to capture the surface's perceptual slipperiness, hardness, and roughness information, respectively. All three component models are made from data that is recorded from interactions between a handheld tool and the real surface. To facilitate data collection, the real surfaces studied in this Chapter were mounted to acrylic using double-sided tape. Our custom device for recording tooltip speed, contact force, and high-frequency acceleration of a tool was described in Section 2.2.1. This section details the recording, data processing, and modeling procedures that we use to build the three component models for a given surface.

4.2.1 Modeling Friction

Friction Data Recording

Ten seconds of data were recorded during an interaction with the physical surface. The experimenter dragged the tool across the surface in a fluid circular motion while the tool's position, orientation, and force were recorded using the tool's internal sensors described in Section 2.2.1.

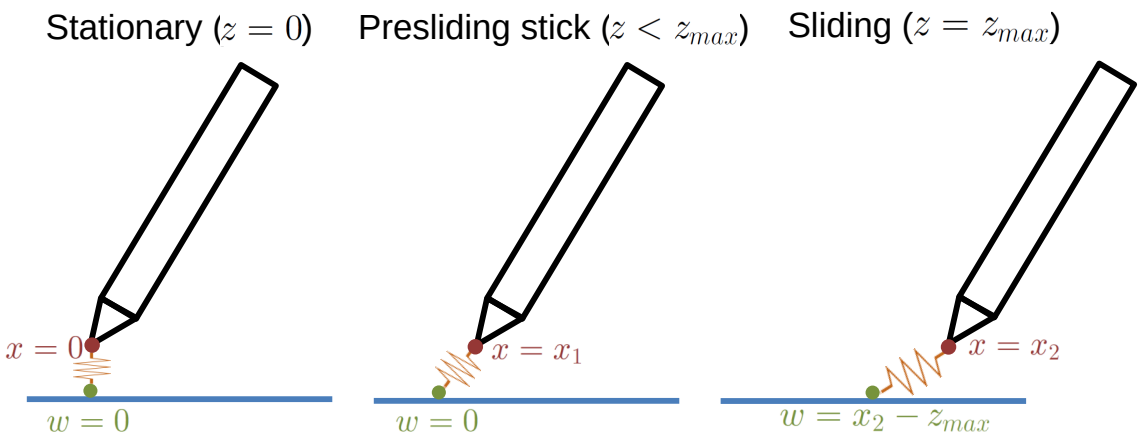

Figure 4.1: Representation of a tool dragging along a surface in the Dahl friction model. The tool and a point on the surface are modeled as being connected by a spring.

Due to the large difference in the sampling rates of the force and position data (10,000 Hz vs. 240 Hz), upsampling the position data does not allow for accurate estimation of the normal and tangential vectors for all time. This interpolation often results in negative estimated friction force values when the direction of travel changes drastically between position measurements. Therefore, we downsampled the three axes of force data to 240 Hz. The tool's position and orientation data were then used to separate the downsampled force measurements into their tangential and normal components. The recordings used to make the friction models in this Chapter are the same as those available in our open-source Penn Haptic Texture Toolkit (HaTT) [22], which is discussed in Appendix A.

Friction Model Creation

The Dahl friction model, which was first presented in [27] and adapted for use in haptic rendering by Hayward et al. [42], represents friction as quasi-static contact bonds

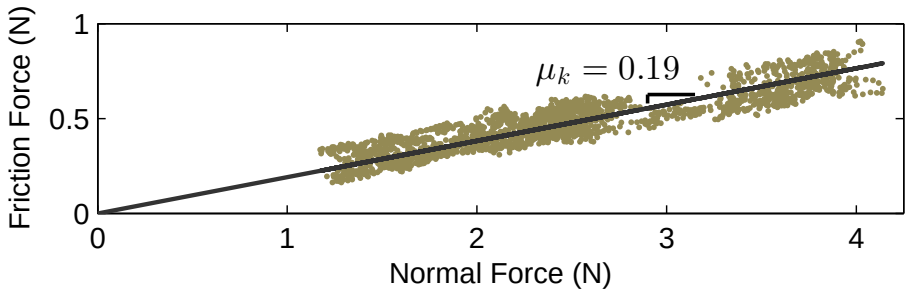

Figure 4.2: Friction data for a sample material, MDF. The friction coefficient is the slope of the best fit line constrained to pass through the origin ($R^2 = 0.84$).

that are continuously formed and then broken. The Dahl friction model beneficially accounts for static friction and the transition to sliding motion. We found that the simplest version of Dahl friction for sliding objects, the stick-slip model, was appropriate for modeling the friction between the recording tool and the surfaces modeled in this Chapter. The stick-slip model has a region of presliding displacement followed by sliding contact with simple Coulomb friction.

The basis for the stick-slip Dahl friction model is shown in Fig. 4.1. The tooltip position x and a point on the surface w are modeled as being connected by a spring. The surface remains fixed as the tool is dragged. The spring connecting x and w stretches until the spring stretch z reaches the maximum value $z_{\max}$. At this stage, the point on the surface moves with the tool to keep the spring stretch constant at $z_{\max}$, and Coulomb friction is followed.

Coulomb friction dictates that the tangential friction force F_f is proportional to the normal force F_n via $F_f = \mu_k F_n$. A line through the origin was fit to the plot of the recorded tangential versus normal force, as shown in Fig. 4.2. The slope of the

line is the kinetic friction coefficient, μ_k.

4.2.2 Modeling Tapping Transients

Tapping Data Recording

The tapping model presented in this Chapter is adapted from the event-based feedback presented in [62]. Ten seconds of position, force, and acceleration data were recorded while the experimenter used the recording device to tap on the selected surface. The experimenter varied the speed at which the contact occurred to capture a wide range of incoming speeds during each recording.

The three recorded acceleration signals from the sensor's three axes were mapped onto a single axis using the DFT321 algorithm presented in [66], which preserves the spectral and temporal properties of the three-axis signals. This mapping is possible because humans cannot determine the direction of vibrations at high frequencies [9]. The combined acceleration was high-pass filtered at 20 Hz to remove the effects of gravity and purposeful human motion. The tool's position data were upsampled to 10 kHz, and the speed of the tool just prior to tapping was calculated using the discrete-time derivative of the upsampled position data.

Tapping Model Creation

The combined acceleration signal was manually partitioned into the individual taps, which were aligned temporally. Each individual tap was labeled with its incoming

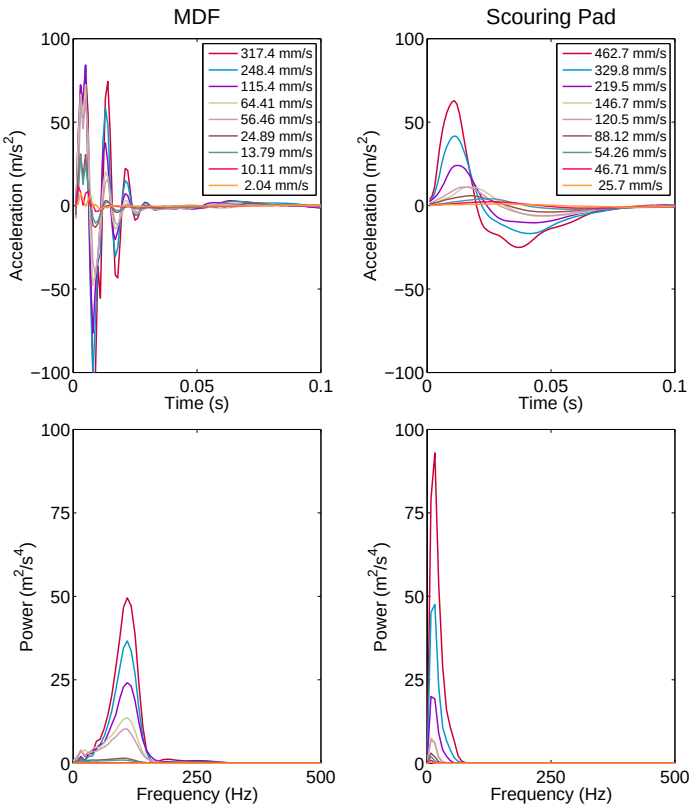

Figure 4.3: Recorded tapping transient accelerations for two sample materials. The amplitude of the vibrations depends on the incoming speed of the tool. The spectral content of the vibrations depends on the hardness of the surface.

speed. Following [62], we saved 100 ms of data after contact. Sample sets of taps for two different materials are shown in Fig. 4.3. The tap response differs significantly between the two materials. The taps from the harder surface (MDF) are of higher frequency and amplitude. The taps from the softer surface (scouring pad) are slower acting and lower power.

4.2.3 Modeling Texture

Our texture modeling methods were presented in Chapter 2. A summary of the data recording and modeling steps are included here for completeness.

Texture Data Recording

Ten seconds of data were recorded for each texture while the experimenter dragged the recording device along the surface using natural and fluid motions. The power and frequency content of the acceleration signals depend greatly on the normal force and scanning speed used during data capture. Therefore, the experimenter used a wide range of forces and speeds to fully capture the behavior of the tool-surface system under many different interaction conditions.

After the data were recorded, they were processed into measurements of normal force, friction force, tooltip speed, and combined acceleration signal as discussed in [26]. The three acceleration signals were combined into a single signal using the DFT321 algorithm [66], and the combined signal was high-pass filtered at 20 Hz to remove the effects of gravity and purposeful human motion. The speed and force signals were low-pass filtered at 100 Hz to remove noise. A sample processed data recording for MDF is shown in Fig. 4.4. The texture recordings used to create the models in this Chapter are the texture data sets available in HaTT [22], which is discussed in Appendix A.

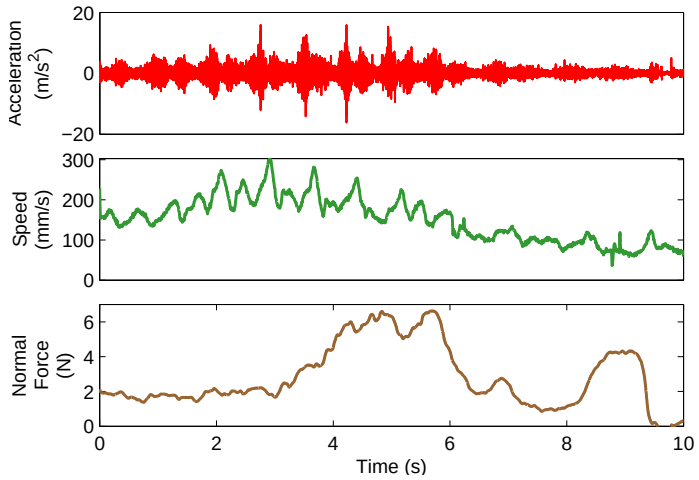

Figure 4.4: Recorded tool acceleration, speed, and normal force signals from dragging the recording tool across MDF in a circular motion. The accelerations change as a function of the scanning speed and normal force used during the interaction.

Texture Model Creation

Model Structure: In this Chapter, as in past work [21, 25, 26, 95, 96], we borrow from the principles of time series analysis and model the acceleration signal using an autoregressive (AR) model structure, an all-pole infinite impulse response (IIR) filter commonly used in speech processing [16]. An AR model is appropriate for modeling our acceleration signal due to the similarities between an auditory signal and the acceleration waveforms recorded here. Defined by its coefficients and variance, an AR model calculates the system's next output value as a linear combination of the previous outputs.

A necessary condition for the AR model structure is strong stationarity, which requires that the behavior of the signal does not change over time. Acceleration signals recorded during natural interactions violate this condition, and therefore we

cannot model the entire signal as a single autoregressive model. Instead, we segment the acceleration signal and model it as a piecewise autoregressive process. We use the Auto-PARM segmentation algorithm [28] to parse the acceleration signal into segments in which the signal's spectral characteristics are approximately constant.

Modeling Process: An AR model was created for each segment, and this model is labeled with the median force and median speed that were used during recording of that segment. The individual AR models for a single material are stored in a Delaunay triangulation with a single AR model at each triangle vertex.

A haptic loop rate of 1000 Hz is required when rendering the textures in a virtual environment created using OpenHaptics and OpenGL, as was done in [21]. Since the acceleration data was recorded at 10 kHz, the AR models must be downsampled by a factor of ten. The model downsampling process creates an equivalent model at 1000 Hz by using a zero-order hold on the inputs, consequently adding zeros to the model. The addition of these zeros transforms the AR models into autoregressive moving-average (ARMA) models, where the output depends on the history of both the past outputs and the past inputs. An ARMA model corresponds to the following discrete-time transfer function:

$$H(z) = \frac{\sum_{k=0}^{q} C_k z^{-k}}{1 - \sum_{k=1}^{p} A_k z^{-k}} \tag{4.2.1}$$

where p is the AR model order, q is the MA model order, and A_k and C_k are the kth AR and MA coefficients, respectively. Furthermore, when the model is resampled,

the spectral density must remain constant. Therefore, the model variance must be scaled by the ratio of the sampling rates:

$$var_2 = \frac{F_{s,2}}{F_{s,1}}(var_1) \tag{4.2.2}$$

where $F_{s,1}$ and $F_{s,2}$ are the original and new sampling rates, respectively.

The texture models created using this process are available in HaTT [22], which is discussed in Appendix A.

4.3 Surface Rendering

This Section discusses our methods for rendering virtual haptic surfaces using the three model components described in Section 4.2. Each component is rendered separately and added to the output of the other components to create the full feel of the surface. The rendering methods presented in this Section build on those described in [26].

During rendering, the Omni measures the user's position and outputs a force. When the user is touching a virtual object, the endpoint position of the haptic device is referred to as the Haptic Interface Point (HIP), and the corresponding point on the surface of the object is referred to as the Ideal Haptic Interface Point (IHIP) [101]. The virtual surface is displayed using a Hooke's law relationship so that the normal force is proportional to the user's penetration depth into the object (the distance between the HIP and the IHIP) with a gain of 0.25 N/mm. This gain was chosen to

allow all surfaces to have the same underlying stiffness; the perceived hardness was then changed between surfaces by the tapping transients alone. A higher gain left the softer surfaces feeling too hard. Furthermore, we wanted subjects to use a large range of normal forces when interacting with the virtual surface without saturating the Omni's force output. Because the surface was rendered using a Hooke's law relationship, the force resolution is proportional to the Omni's position resolution. When the stiffness value is high, only a small penetration depth is needed to reach the maximum force, making it difficult for users to apply small forces.

4.3.1 Rendering Friction

The surface friction was rendered using the stick-slip Dahl friction model presented in Section 4.2.1. The HIP x and the IHIP w are connected by a virtual spring. The first stage of spring stretch represents the presliding stick shown in Fig. 4.1. The second stage of spring stretch, the sliding region, follows simple Coulomb friction. The location of the adhesion point on the surface is calculated as:

$$w_k = \begin{cases} x_k - \text{sgn}(x_k - x_{k-1}) z_{\max} & \text{if } |x_k - w_{k-1}| > z_{\max} \\ w_{k-1} & \text{otherwise} \end{cases} \tag{4.3.1}$$

where w_{k-1} is the previous location of the adhesion point, x_k is the current location of the tool, and x_{k-1} is the previous location of the tool. The value of $z_{\max}$ was empirically tuned to ensure stability of the system. Surfaces with higher friction required a larger value for $z_{\max}$ than surfaces with low friction, leading us to choose

$z_{\max}$ to be proportional to the kinetic friction coefficient ($z_{\max} = 4\,\text{mm} \cdot \mu_k$). The friction force is then calculated as:

$$\begin{aligned} z_k &= x_k - w_k \\ F_f &= \mu_k F_n \left(\frac{z_k}{z_{\max}} \right) \end{aligned} \tag{4.3.2}$$

The friction force is rendered using the motors of the Omni and is displayed opposite the direction of motion of the IHIP.

4.3.2 Rendering Tapping

During rendering, the displayed tapping response is a function of the user's incoming contact speed v_{in}, as in [62]. Because the user's mass is constant, the momentum of the Omni-hand system ($p = mv_{\text{in}}$) varies linearly with incoming velocity. Therefore, the amplitude of the transients must also scale with the incoming velocity. This velocity dependence is also shown in the recorded tapping transients in Fig. 4.3. For incoming speeds lower than the minimum recorded tapping speed, the lowest speed transient is scaled by the quantity $v_{\text{in}}/v_{\min}$. For incoming speeds higher than the maximum tapping speed, the highest speed transient is directly played. For intermediate speeds, the transients from the two bounding speeds are interpolated before being played.

Although the tapping transients are recorded as accelerations, they must be rendered as forces to provide the momentum cancellation necessary to elicit the sensations of tapping on a stiff surface. If the tapping transients are instead played as vibrations, such as through a voice-coil actuator, the user's tool continues to penetrate

the surface, creating the sensation of a soft surface. Therefore, to play the tapping vibrations as a force, we first scaled the signal by the effective mass of the handle and the user's hand (0.05 kg, as determined in [21]) before playing it through the motors of the Omni in the direction of the surface normal. Although adding the tapping force to the stiffness response momentarily exceeds the maximum steady-state force output of the device, the duration of the high force is sufficiently small (< 0.1 s) not to overheat the motors.

4.3.3 Rendering Texture

This Section describes the methods for synthesizing and rendering a texture vibration signal using the models created as described in Section 4.2.3. This Section extends the texture rendering methods that were presented in [21] to include texture signal output through a dedicated actuator (a Haptuator) instead of through the motors of the Omni.

Texture Signal Generation

In real time we generate a texture vibration signal that varies according to the interaction conditions. Measurements of the user's current normal force and tangential speed are used to interpolate between models in the Delaunay triangulation. To ensure stability, we first convert the ARMA coefficients into line spectral frequencies (LSF) before interpolation, as described in [60]. After interpolation, the LSFs are

converted back into ARMA coefficients. New coefficients and variance values are calculated at a rate of 1000 Hz.

A white Gaussian excitation signal with power equal to the interpolated variance is generated and used to drive the interpolated model. The excitation signal's history and a history of the previous vibration outputs are used to calculate the new vibration output at a rate of 1000 Hz.

Hardware

In our initial work on this topic [21], the texture vibrations were displayed using the motors of the Omni primarily to simplify the implementation of our open-source rendering methods by others. However, displaying the vibrations through the Omni motors distorts the signal significantly because the connection from motor to hand is not rigid [79]. Therefore, we updated our rendering system to output the texture vibrations through a dedicated voice-coil actuator, a Haptuator (Tactile Labs TL-002-14R); this model is an updated version of the model used in our TexturePad system [26]. We first considered putting the Haptuator within the body of a custom-made Omni handle, but this location requires the Haptuator to be placed behind the user's grip. The vibrations then did not seem to emanate from tooltip contact, degrading the user experience. Therefore, we placed the Haptuator at the tip of the Omni handle, as shown in Fig. 4.5. We found that aligning the Haptuator axially to the handle felt better than orthogonal to the handle, presumably because of the

Figure 4.5: Omni with Haptuator attached to the stylus tip.

mechanical properties of the gimbal. The Haptuator was attached with a zip tie.

The texture vibrations are scaled and output through an analog output pin on a Sensoray 626 PCI card at a rate of 1000 Hz. This output voltage is then passed through a linear current amplifier with a gain of 1 A/V to drive the Haptuator.

4.4 Choosing Distinct Surfaces

Building from the database of friction and texture models in [22], we made a complete set of 100 surface models including friction, tapping transient, and texture components. This Section discusses the steps we used to select a set of fifteen perceptually distinct surfaces for use in the human subject studies presented in the following Sections. We chose the surfaces by examining physical properties that correspond to perceived slipperiness, hardness, and roughness.

4.4.1 Friction Estimation

As described in Section 4.2.1, a Dahl friction model was fit to the data recorded for each surface. The resulting kinetic friction coefficient was used to represent the slipperiness of each surface.

4.4.2 Tapping Transient Estimation

The perceived hardness of each surface was estimated by fitting a line to a plot of measured peak normal force at tap versus incoming speed, using the surface tapping responses described in Section 4.2.2. The slope of this line, which we refer to as the tap response factor, was shown to be a valid measure of the primary trend of the dynamics of tapping transients in [32].

4.4.3 Roughness Estimation

We created an indicator for the perceived roughness of the surfaces by synthesizing vibration signals for all 100 textures using a representative force-speed trajectory that spans the area of force-speed space typically explored by humans. The total power of the resulting vibration signal was calculated; this value has been shown to correspond to the roughness of the surface [10, 11, 58, 119].

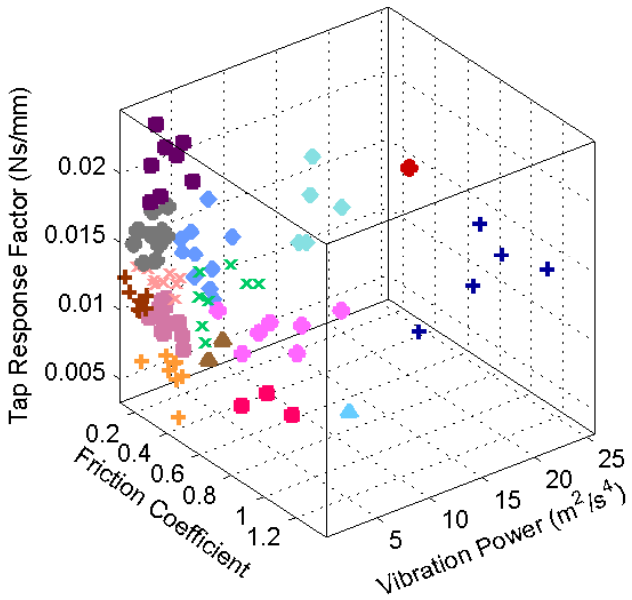

Figure 4.6: Clusters of surfaces based on kinetic friction coefficient, tap response factor, and power of texture vibrations. A subset of surfaces that spans the perceptual space represented by the original 100 surfaces was chosen by selecting one surface from each cluster.

4.4.4 Surface Clustering

We plotted the 100 surfaces in the three-dimensional space defined by these three measurements, as shown in Fig. 4.6. The studies presented in Sections 4.5 and 4.6 required a subset of surfaces that are distinct and span the perceptual space. To choose this subset, we performed hierarchical clustering on the z-scores of the three parameters (friction coefficient, tap response factor, and vibration power). An analysis of the scree plot of the distances to cluster centroid indicated that about fifteen clusters were needed to form distinct groups of surfaces and capture the variation between groups. Fig. 4.6 shows the resulting clustering for fifteen clusters. We wanted the

Figure 4.7: The fifteen surfaces chosen from the cluster analysis to span the perceptual space covered by the original 100 surfaces.

materials to span the perceptual space, so we chose one surface from each cluster. The fifteen chosen surfaces are shown in Fig. 4.7.

4.5 Human Subject Study: Responsiveness to User Force and Speed

This Section describes the human-subject experiment we ran to assess how the texture vibrations' responsiveness to user force and speed affect the realism of the virtual textures. All procedures were approved by the Penn IRB under protocol 820685. Fifteen subjects were recruited for the experiment (eight males and seven females). Thirteen subjects were right-handed, and two subjects were left-handed. Six subjects had no experience with haptic devices, eight subjects had limited experience, and one subject had extensive experience.

Figure 4.8: Experimental set-up. The subject sat at a table in front of a computer and wore headphones playing pink noise to mask auditory cues. The subject used an Omni to feel the virtual surfaces and an adapted Omni handle to feel the real surfaces.

4.5.1 Experimental Set-up

The subject sat at a desk in front of a computer throughout the study and wore headphones playing white noise to mask outside sounds, as shown in Fig. 4.8. This intervention ensured that the subject used only haptic, and not auditory, cues during the study. The subject interacted with the physical surfaces using a detached Omni handle that was modified to have a 3.175-mm-diameter stainless steel hemispherical tooltip, chosen to match the tooltip used during data recording. Each surface was mounted on a 10 cm×10 cm piece of acrylic using double-sided tape. During the experiment, the physical surface was placed in a stage that holds the surface stationary

above the table so the real and virtual surfaces are presented at the same height. The corresponding virtual surface was represented as a horizontal plane to match the size and orientation of the physical surface, and an image of the surface was displayed on the screen, as shown in Fig. 4.8. The experimenter monitored the subject's hand during the experiment to ensure that it did not contact the surface or the Haptuator. Subjects were instructed to limit their normal force to 2.75 N when touching the virtual surface to avoid saturating the motors of the Omni, and a visual indicator was activated if the subject exceeded this force threshold.

4.5.2 Experimental Procedure

Before beginning the study, subjects were allowed to feel all of the real surfaces with the tool for as long as desired. This practice ensured that the subjects were familiar with the surfaces and the range of haptic sensations they would feel during the study. Subjects were also given time to practice using the Omni by exploring a simple haptic environment consisting of a sphere and cube inside of a box.

In the study, subjects rated the similarity between each real surface and four virtual versions of that surface. The four virtual surfaces were created by altering the texture vibrations' responsiveness to user force and speed. The control surface (Condition FS) was created by allowing the texture vibrations to respond to both the user's normal force and tangential speed. The texture vibrations for the second surface (Condition F) responded only to normal force, and speed was assumed to be a

Table 4.1: Rendering Conditions

	Normal Force	**Tangential Speed**
Condition FS	user force	user speed
Condition F	user force	S_c=130 mm/s
Condition S	F_c=1.2 N	user speed
Condition ϕ	F_c=1.2 N	S_c=130 mm/s

constant value. The texture vibrations for the third surface (Condition S) responded only to tangential speed, and normal force was assumed to be a constant value. The texture vibrations for the fourth surface (Condition ϕ) did not respond to the user's normal force or tangential speed; they were generated assuming a constant force and speed. For Conditions F and ϕ, speeds less than 1.0 mm/s were rounded to zero so that no vibrations were played after the user stopped moving. A summary of the experimental conditions is shown in Table 4.1. The tapping transient and friction rendering were the same in all conditions.

The constant force and constant speed values were chosen using the modeled forces and speeds for the fifteen textures in this study because they are representative of the forces and speeds humans use when interacting with the physical surfaces. The mean of the modeled force and mean of the modeled speed were calculated across all textures, excluding models with a force greater than the saturation force (2.75 N). The constant force value for the rendering conditions was set equal to the mean modeled force ($F_c = 1.2$ N), and the constant speed value was set equal to the mean modeled speed ($S_c = 130$ mm/s).

Samples of the output signals for the four rendering conditions are shown in

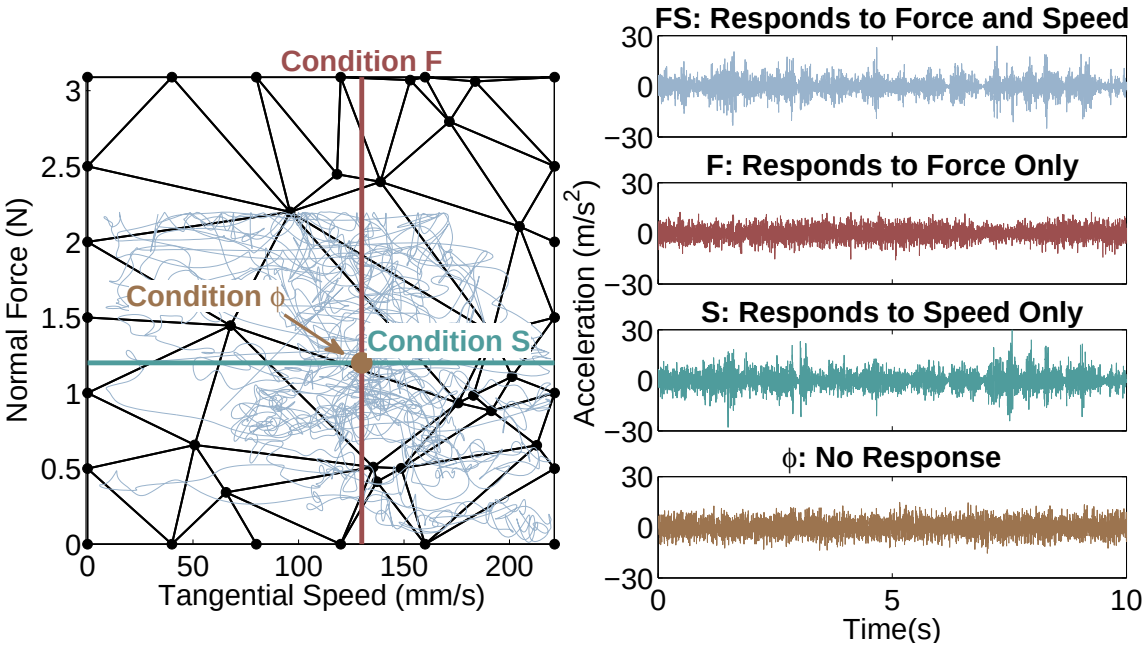

Figure 4.9: Sample texture vibration signals that were synthesized for the four rendering conditions using the Nylon Mesh texture model. The force-speed trajectory was chosen to span the typical force-speed space explored by subjects when feeling the surfaces.

Fig. 4.9 for the Nylon Mesh material. A sample force-speed trajectory was chosen to span the typical forces and speeds used by humans when dragging a tool across a surface. The force-speed trajectory, which is shown overlaid on the Delaunay triangulation of models for Nylon Mesh, remains inside the convex hull at all times. The control texture signal (Condition FS), which responded to both user force and speed, was synthesized using this force-speed trajectory. The signal for Condition F was synthesized using the force measurements of the trajectory with the speed set to a constant value; this modified trajectory lies on the line of constant speed shown in Fig. 4.9. The signal for Condition S was synthesized using the speed measurements of the trajectory with the force set to a constant value; this modified trajectory lies

on the line of constant force. The signal for the Condition ϕ was synthesized using a constant force-speed signal; this modified trajectory is shown as a single point of constant force and speed.

The subject was presented with one real surface and one of its four corresponding virtual surfaces in each trial. The real surfaces were presented in randomized order, as were the four rendering conditions within each surface. The subject rated the similarity between a real surface and all four of its virtual versions before moving on to the next real surface. Subjects were asked to rate the similarity of the virtual surface to the corresponding real surface by placing a mark on a visual analog scale from "completely different" to "completely the same". Study data were collected and managed using Research Electronic Data Capture (REDCap) tools hosted by the University of Pennsylvania [41].

To best capture the effect of force and speed responsiveness on the perceived similarity, we required subjects to use a range of normal forces and tangential speeds when exploring each virtual surface before they were allowed to provide a rating for the similarity between the real and virtual surface. The force-speed space was divided into four quadrants: gentle force, slow speed; gentle force, fast speed; firm force, slow speed; firm force, fast speed. The center of the force-speed space was set to (S_c, F_c). Subjects were required to spend at least 5 seconds in each quadrant and were given a visual indicator to display how much time they had spent in each quadrant, as shown in Fig. 4.10. The circle indicators sequentially changed from red to green as

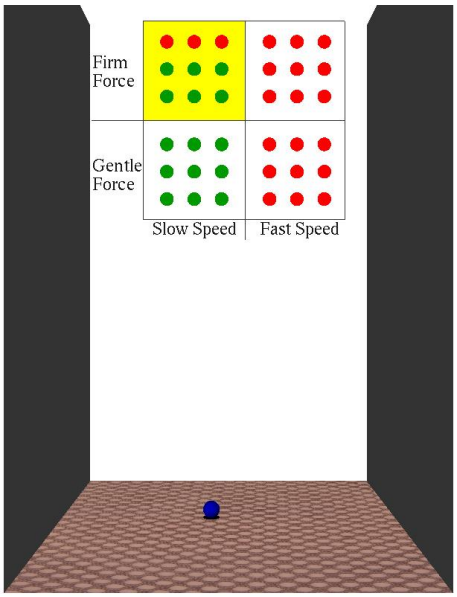

Figure 4.10: Experimental set-up. The subject used a Phantom Omni with Haptuator rigidly attached to the tip to interact with the virtual textures.

the time spent in that quadrant increased. The quadrant currently being explored by the subject was highlighted in yellow.

4.5.3 Results

Each subject's similarity ratings were normalized individually for each material by subtracting the mean of the subject's similarity ratings for that material. This normalization scheme eliminated between-subject and between-material differences without affecting within-material differences. The normalized similarity ratings are shown in Fig. 4.11. A one-way analysis of variance (ANOVA) was performed for each material on the normalized similarity ratings with rendering condition as the factor. A multiple comparison test was conducted on the ANOVA results to determine which conditions, if any, had a significantly different mean from the control condition for

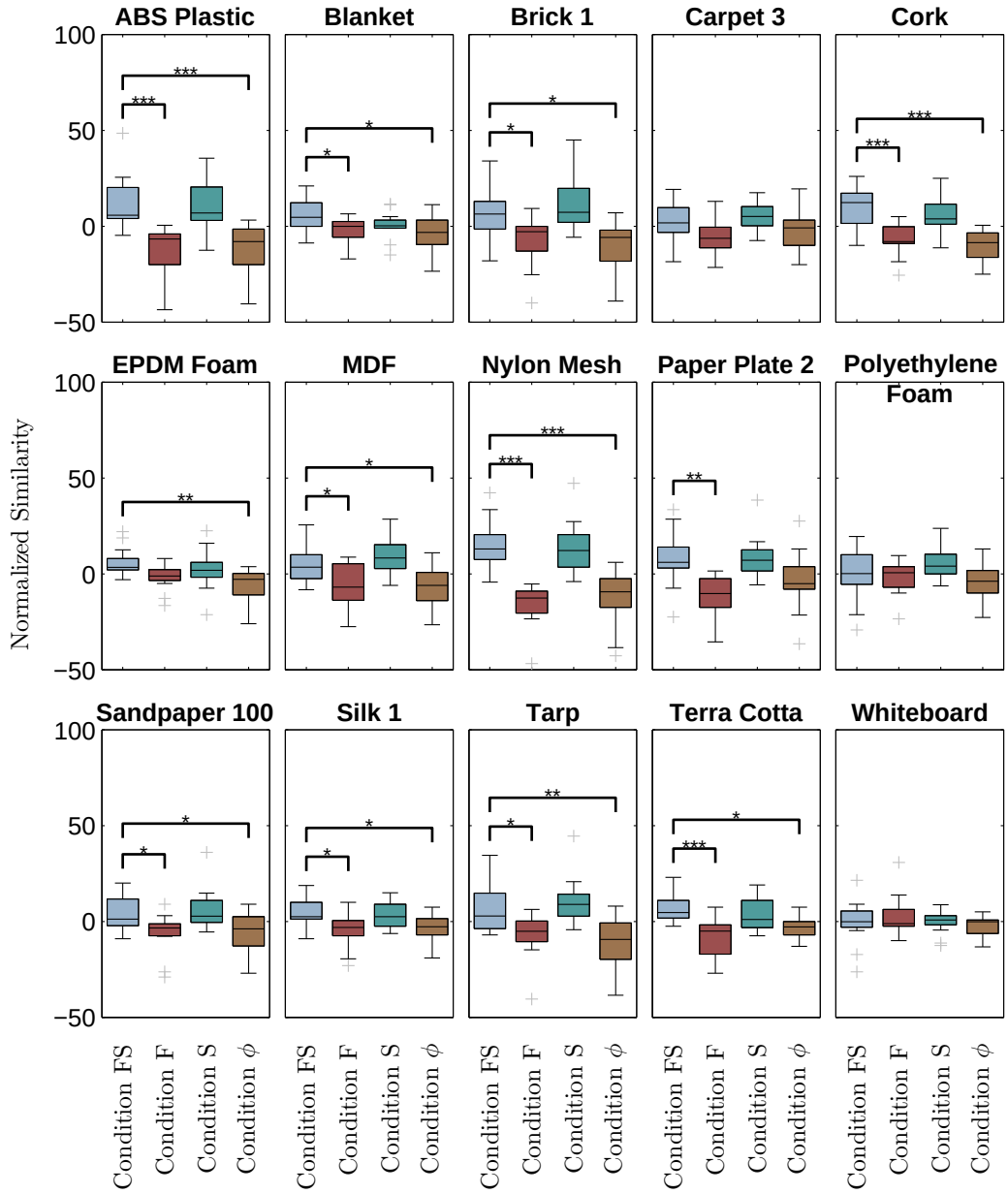

Figure 4.11: Comparison of similarity ratings across the four virtual versions of each surface (Condition FS: responds to force and speed; Condition F: responds to force only; Condition S responds to speed only; Condition ϕ: does not respond to force or speed.). Statistically significant differences in similarity value from the control virtual surfaces are marked with a bar and asterisks ($*** \equiv p \leq 0.001$, $** \equiv p \leq 0.01$, $* \equiv p \leq 0.05$).

each material.

Removing the texture's response to user speed (Condition F) caused a statistically significant decrease in similarity for eleven surfaces: ABS Plastic ($p = 6.12 \times 10^{-5}$), Blanket ($p = 0.049$), Brick 1 (0.039), Cork ($p = 9.28 \times 10^{-5}$), MDF ($p = 0.010$), Nylon Mesh ($p = 1.83 \times 10^{-7}$), Paper Plate 2 ($p = 0.0011$), Sandpaper 100 ($p = 0.039$), Silk 1 ($p = 0.018$), Tarp ($p = 0.020$), and Terra Cotta ($p = 5.92 \times 10^{-5}$). Conversely, removing the texture's response to user force (Condition S) did not cause a statistically significant change in similarity for any of the surfaces. Removing the texture's response to both user force and user speed (Condition ϕ) caused a statistically significant decrease in similarity for eleven surfaces: ABS Plastic ($p = 8.75 \times 10^{-5}$), Blanket ($p = 0.017$), Brick 1 ($p = 0.010$), Cork ($p = 5.45 \times 10^{-6}$), EPDM Foam ($p = 0.0024$), MDF ($p = 0.021$), Nylon Mesh ($p = 2.56 \times 10^{-6}$), Sandpaper 100 ($p = 0.047$), Silk 1 ($p = 0.029$), Tarp ($p = 0.0024$), and Terra Cotta ($p = 0.012$). However, there was not a statistically significant difference in similarity for surfaces rendered without speed responsiveness (Condition F) and surfaces rendered without both force and speed responsiveness (Condition ϕ).

To compare the effects of removing force or speed responsiveness on the behavior of texture vibration signals, a set of signals was synthesized using the force-speed trajectory shown in Fig. 4.9. Four signals were synthesized for each texture following the four rendering conditions. Both power and frequency spectra are important cues in determining how haptic textures feel. Therefore, both should be taken into account

when comparing the effects of force and speed response on the behavior of texture vibration signals. The signals for Conditions F, S, and ϕ were compared to the control signal (Condition FS) to determine how the power and spectrum of the signals change when responsiveness was removed.

A windowed discrete Fourier transform was calculated for each signal using non-overlapping windows of 32 datapoints, and the power in each window was calculated from the DFT. The change in the signal's power over time, which was calculated as the standard deviation of the windowed power, is a measure of the power's dependence on changing user inputs. In Condition F, for example, changes in the signal's power over time are directly caused by changes in user force. We calculated the change in power variation ($\widehat{\mathrm{P}}$) between Condition FS and Conditions F, S, and ϕ to determine the effect of removing responsiveness to user input on the signal's power:

$$\widehat{\mathrm{P}}_{\mathrm{x}} = \mathrm{std}(\mathrm{P}_{\mathrm{FS}}) - \mathrm{std}(\mathrm{P}_{\mathrm{x}}) \tag{4.5.1}$$

where P_{FS} is the windowed power of the control signal and P_{x} is the windowed power of the signal for the condition of interest. This subtraction normalizes the power variation so $\widehat{\mathrm{P}}_{\mathrm{x}}$ is not affected by differences in overall signal power. The difference between the two signals' power variation estimates the amount of variation in power that is lost when the signal's response to user force and/or speed is removed. The decrease in similarity was plotted against the change in power variation for the three test conditions and a line of best fit was calculated. The slopes of the best fit lines

Table 4.2: Comparison of Similarity to Texture Signal Behavior

Condition	$\frac{\Delta \text{Sim}}{\Delta \widehat{\text{P}}_\text{x}}$ (R^2)	$\frac{\Delta \text{Sim}}{\Delta \widehat{\text{SC}}_\text{x}}$ (R^2)	$\frac{\Delta \text{Sim}}{\Delta \text{SD}_\text{x}}$ (R^2)
Condition F	875.5 *(0.33)*	3.46 *(0.22)*	2.23 *(0.38)*
Condition S	1093.5 *(0.30)*	1.24 *(0.28)*	1.59 *(0.32)*
Condition ϕ	960.4 *(0.57)*	3.20 *(0.31)*	4.70 *(0.65)*

are shown in the first column in Table 4.2.

To compare the frequency spectra of the signals over time, the spectral centroid was calculated for each window of the signal's DFT using the equation:

$$\text{SC}_j = \frac{\sum_k X_j(f_k)^2 f_k}{\sum_k X_j(f_k)^2} \tag{4.5.2}$$

where $X_j(f_k)$ is the amplitude of the DFT in the jth window at frequency f_k. The change in the signal's spectral centroid over time, which was calculated as the standard deviation of the windowed spectral centroid, is a measure of the spectrum's dependence on changing user inputs. We calculated the change in spectral centroid variation ($\widehat{\text{SC}}$) between Condition FS and Conditions F, S, and ϕ to determine the effect of the user input on the signal's spectral centroid:

$$\widehat{\text{SC}}_\text{x} = \text{std}(\text{SC}_\text{FS}) - \text{std}(\text{SC}_\text{x}) \tag{4.5.3}$$

where SC_FS is the windowed spectral centroid of the control signal and SC_x is the windowed spectral centroid of the signal for the condition of interest. The difference between the two signals' spectral centroid variations estimates the amount of spectral variation that is lost when the signal's response to user force and/or speed is removed. The decrease in similarity was plotted against the change in spectral centroid variation

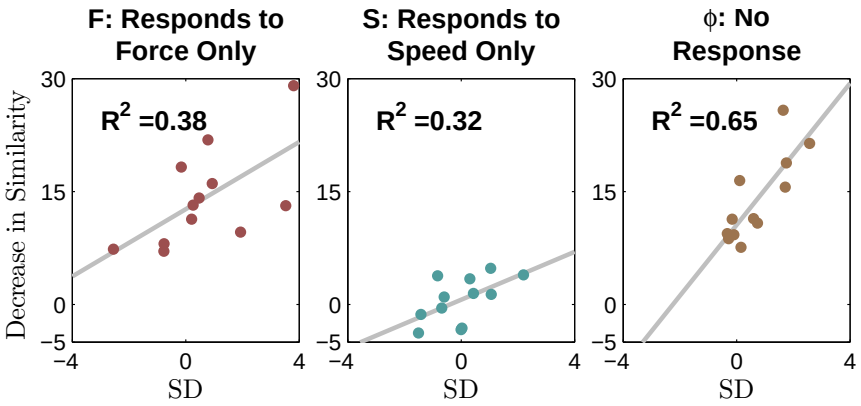

Figure 4.12: Comparison of the change in similarity between the control surface and the three experimental conditions to the conditions' spectral difference metric.

for the three test conditions and a line of best fit was calculated. The slopes of the best fit lines are shown in the second column in Table 4.2.

We proposed the spectral difference (SD) metric in [25] to compare two signals with equal emphasis on spectral shape and power. We use a modified version of this metric to compare Condition FS to Conditions F, S, and ϕ:

$$\mathrm{SD}_{\mathrm{x}} = \frac{\widehat{\mathrm{P}}_{\mathrm{x}}}{\mathrm{std}(\widehat{\mathrm{P}}_{\mathrm{x}})} + \frac{\widehat{\mathrm{SC}}_{\mathrm{x}}}{\mathrm{std}(\widehat{\mathrm{SC}}_{\mathrm{x}})} \tag{4.5.4}$$

Normalizing $\widehat{\mathrm{P}}_{\mathrm{x}}$ and $\widehat{\mathrm{SC}}_{\mathrm{x}}$ yields a dimensionless metric that weights the two factors equally. The SD estimates the amount of power and spectral variation that is lost when the signal's response to user force and/or speed is removed. The decrease in similarity is plotted against the spectral difference for the three conditions in Fig. 4.12, and a line of best fit was calculated for each test condition. The slopes of the best fit lines are shown in the third column in Table 4.2.

4.5.4 Discussion

By evaluating the change in similarity between real and virtual textures caused by the removal of force and/or speed responsiveness, this study elucidated the conditions necessary to create realistic haptic textures. The similarity ratings shown in Fig. 4.11 indicate that allowing texture vibrations to respond to user speed alone is sufficient for creating realistic virtual textures. Removing either speed responsiveness or both force and speed responsiveness caused a statistically significant decrease in similarity rating compared to the control condition for twelve of the fifteen textures. However, removing force responsiveness alone did not cause a statistically significant change in real-virtual similarity for any of the textures. Furthermore, removing both force and speed responsiveness was not significantly different from removing speed responsiveness alone. Therefore, removing force responsiveness did not significantly alter how realistic the virtual textures felt in this study. Although the addition of force responsiveness caused an insignificant increase in similarity for seven of the textures (ABS Plastic, Blanket, Cork, EPDM Foam, Nylon Mesh, Silk 1, and Terra Cotta), it may not be feasible or practical to add force responsiveness due to limitations in texture recording, modeling, or rendering. On the other hand, estimating user speed from measurements of the user's position over time is easier both computationally and in terms of hardware, and it seems to be sufficient for rendering many haptic texture vibrations.

Some textures, however, did not have a significant change in similarity when speed

responsiveness was removed, which can be explained by physical characteristics of the textures or limitations of the models. For example, we believe the differences in similarity for Whiteboard are not significant because the vibrations produced from dragging across its smooth surface are small and do not vary greatly with force and speed. Therefore, subjects likely could not distinguish the subtle changes in the signal for the four rendering conditions. The differences in similarity for Carpet 3 were also not statistically significant due to properties of the surface itself. This sample is comprised of a weave of strands attached to a base material. When dragging a tool along the carpet at low forces, the tool remains on top of the woven material, which produces texture vibrations that are well captured by our modeling and rendering methods. However, if higher forces are applied, the tool can get stuck in the material, eliciting impulse forces that cause the tool to temporarily stop. This behavior is not captured by our current methods, which we believe greatly affected the similarity ratings for this material. Several subjects commented on this lack of behavior of the virtual surface as being a detriment to similarity. Since we could not accurately match this behavior, the texture models for higher forces were less realistic than lower force models, which caused confusion in the similarity renderings. Lastly, we believe the differences in similarity for Polyethylene Foam were not significant due to the range of the texture's modeled forces and speeds. There were few models above the constant force and constant speed values used in rendering Conditions F, S, and ϕ. Therefore, the behavior of the signal did not change significantly if the subject used

high forces and/or high speeds. Since the overall variation in the displayed signal was less than expected, none of the similarities were significantly different between the rendering conditions. This result highlights the need to record sufficient data to model the haptic texture at the interaction conditions the user will employ when feeling the virtual textures.

The sample signals shown for the four rendering conditions in Fig. 4.9 support our result that responding to user speed is more important than responding to user force. The signal that responds to force alone was fairly uniform across the time window except for a slight decrease in the power of the signal at about $t = 7$ s. In contrast, the signal that responds to speed alone appears more similar to the control signal, which responds to both force and speed. It exhibits the same pulse behavior of the control signal, and the power varies greatly over the time window. As expected, the signal that did not respond to either force or speed was uniform for all 10 seconds. These observations match our result from the rated similarity values, which showed that responding to user speed is more important than responding to user force.

Although our results show that allowing texture vibrations to respond to user speed is essential for increasing realism, the relationship between the change in similarity and changes in the behavior of the texture signals depends on which response component is removed. Table 4.2 illustrates how changes in similarity reflect changes in power and spectral behavior of signals generated for the different rendering conditions. In the first column of this table, we compared the power variation in a repre-

sentative synthesized texture signal for Conditions F, S, and ϕ to the power variation in the signal synthesized for the control Condition FS. A higher slope ($\Delta\text{Sim}/\Delta\widehat{\text{P}}$) signifies that the perceived similarity was more sensitive to the changes in power caused by the response component(s) that was removed. The slope was highest for Condition S, indicating that the perceived similarity was most sensitive to changes in power caused by the signal's response to user force. However, $\Delta\text{Sim}/\Delta\widehat{\text{P}}$ was similar for all three test conditions, which suggests that the response of the signal's power to both user force and speed was important to the realism of the virtual textures.

In the second column of Table 4.2, we compared the spectral centroid variation in a representative synthesized texture signal for Conditions F, S, and ϕ to the spectral centroid variation in the signal synthesized for the control Condition FS. The calculated slope ($\Delta\text{Sim}/\Delta\widehat{\text{SC}}$) was much higher for Conditions F and ϕ than for Condition S. This disparity indicates that the perceived similarity was most sensitive to changes in the signal's spectrum caused by the signal's response to user speed. Conversely, the perceived similarity was not highly sensitive to changes in the signal's spectrum caused by response to user force. This trend is also seen in the spectral difference metric, which equally considered the contribution of both the power and frequency spectra to the change in realism, as shown in Fig. 4.12 and the third column of Table 4.2. Although the spectral difference is on the same order of magnitude for all three conditions, changes in spectral difference when speed or both force and speed responsiveness are removed correlate to higher changes in realism than when force

responsiveness alone is removed. This result supports our previous conclusion that responding to user speed is more important to realism than responding to user force.

While we believe this result reflects a fundamental difference in the importance of force and speed in rendering haptic textures, the result may have been affected by the limited range of normal forces subjects could use in the study. However, only three of the fifteen subjects activated the high force indicator, which shows that the subjects naturally used forces lower than the force limit (2.75 N). Furthermore, from the study data captured in [25], we found that the average force applied by subjects when dragging a tool across a texture was 1.1 N. Therefore, the forces that subjects were able to use in this current study were within the range of normal forces we have found that humans typically use and likely did not have a large effect on the results.

The results of this study for textures felt through a tool are in contrast with past work in modeling textures felt with a bare finger, which showed that a texture can be represented by the spatial spectrogram of tangential forces felt during dragging [113]. To render texture signals at speeds other than the speed used during data recording, Wiertlewski et al. sped up or slowed down the rate at which the time-domain force signal was played. However, our results indicate that the spatial spectrogram approach cannot be used to represent textures that are felt through a tool because changes in tool speed alter both the power and frequency spectrum of the vibration signal. This variation is due to the behavior of the tooltip when it contacts the surface features during dragging [77]. When tool speed is increased, the tooltip hits each sur-

face feature with more momentum, causing a larger impulse and resulting in higher power vibrations. Therefore, for tool-based texture rendering, data must be recorded at many different tangential speeds to best capture the full feeling of the texture.

The subjects in this study were predominantly untrained and inexperienced users, which matches a large portion of the population of potential users of haptic technology. Although experts trained in haptic discrimination tasks may be more sensitive to how texture signals change with force, this study showed there is not a significant change in realism when force responsiveness was removed. However, subjects rated the textures as significantly less realistic when the vibrations did not respond to user speed. Therefore, for the average user, allowing texture vibrations to respond to user speed and not user force should be sufficient to create realistic haptic textures.

This study focused on the overall realism of the virtual textures and does not consider the perceptual qualities (e.g., roughness, hardness, and slipperiness) that may be affected by the removal of force and/or speed responsiveness. For example, past research has shown that the power of the texture signal determines the perceived roughness of both virtual [104] and real [119] textures. In this study the changes in signal power had the largest effect on realism when force responsiveness was removed, which may indicate that changes in perceived roughness caused the decreases in realism observed in Condition S. Further research is needed to determine the full effect of the textures' force and speed responsiveness on the separate perceptual qualities.

4.6 Human Subject Study: Friction, Tapping and Texture

This Section describes the human-subject experiment we ran to assess the realism of our haptic virtual surfaces. We also sought to determine the perceptual importance and ecological validity of displaying friction, tapping, and texture information on the virtual surfaces. Described in [68], ecological validity is the correlation between a cue and a property of the world. We wanted to determine how the realism of the virtual surfaces is affected by the three separate modeling components, and how this relationship depends on the physical surface parameters.

All procedures were approved by the Penn IRB under protocol 820685. Two separate sets of fifteen subjects were recruited for a total of thirty subjects. Twenty-seven subjects were right-handed, and three subjects were left-handed. Twenty-nine subjects had limited experience with haptic devices, and one subject had extensive experience.

4.6.1 Experimental Set-up

The experimental set-up in this study was the same as that in Section 4.5 and is shown in Fig. 4.8. When interacting with the real or virtual surfaces, subjects were instructed to use any motions they desired including dragging, tapping, and pressing into the surface. To avoid saturating the motors of the Omni, a visual indicator

activated when the subject pressed on the virtual surface with more than 2.75 N of normal force.

4.6.2 Experimental Procedure

There were three phases to the study: adjective ratings of the real surfaces, adjective ratings of the virtual surfaces, and similarity ratings. The first and third phases were completed by one set of 15 subjects. The second phase was completed by the other set of 15 subjects. Before the study began, all subjects practiced with the experimental setup. They were allowed to feel all of the real surfaces for as long as desired using the tool. The subjects were also allowed to practice the motions involved in using the Omni and become familiar with the forces output by the Omni by exploring a simple haptic 3D environment consisting of a box with a stationary sphere and cube rendered using only Hooke's law. Study data were collected and managed using Research Electronic Data Capture (REDCap) tools hosted by the University of Pennsylvania [41].

Real Adjective Ratings

The real surfaces were presented to the subject one at a time in randomized order. The subject was asked to rate each surface on the scales of hard-soft, slippery-not slippery and rough-smooth. These scales were chosen from the list provided in [81] as the psychophysical dimensions applicable to the perception of surfaces through a

tool. The adjective scales were presented in randomized order, and the subject rated all 15 surfaces on a single scale before moving on to the next scale.

Virtual Adjective Ratings

The procedure described above was completed by a second set of 15 subjects. These subjects were asked to rate the 15 virtual surfaces on the same three adjective scales. These virtual surfaces included surface stiffness plus friction, tapping transients, and texture vibrations.

Similarity Ratings

Subjects cannot directly assess the relative importance of the three different haptic modeling components because the information provided by the friction, tapping response, and texture vibrations is processed together and may provide redundant cues. The study in [72], which tested the relative contribution of tactile and auditory information to judgments of roughness, overcame this problem by removing tactile and auditory information one at a time. The perceptual discrepancy between the two feedback components allowed subjects to disambiguate the relative contributions of the tactile and auditory information. We followed a similar perceptual discrepancy paradigm by turning the modeling components off one at a time. This study design created four virtually rendered surfaces for each real surface: the control surface displaying all components, one surface without friction, one surface without tapping transients, and one surface without texture vibrations.

The subject was simultaneously presented with one real surface and one of its four corresponding virtual surfaces. After exploring both surfaces, the subject was asked to rate the similarity between the two surfaces by placing a mark on a scale of "completely different" to "completely the same" on a computer using a mouse. The subject was required to tap on the virtual surface at least five times and drag along the surface for at least 50 cm before providing their rating. The real surfaces were presented in randomized order, and the four virtual surfaces were also presented in randomized order. The subject rated the similarity between a real surface and all four virtual versions of that real surface before moving on to the next real surface.

4.6.3 Results

Adjective Ratings

The adjective ratings were normalized by dividing each rating by the standard deviation of all ratings given by that subject for that scale. The normalized adjective ratings were then averaged across all subjects.

Fig. 4.13 shows the correlation between the average adjective ratings and measured physical parameters calculated in Section 4.4. The average rated slipperiness was moderately inversely correlated with the measured friction coefficient ($R^2 = 0.53$, $p = 2.20 \times 10^{-3}$), and the average rated roughness was highly correlated with the acceleration power ($R^2 = 0.68$, $p = 1.57 \times 10^{-4}$). Although the tap response factor was used to characterize the 100 surfaces in the clustering analysis in Section 4.4,

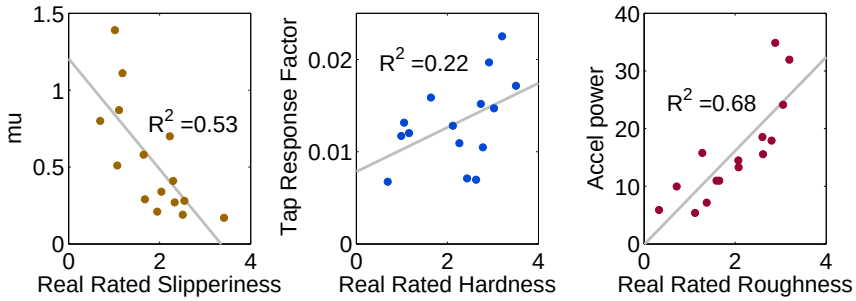

Figure 4.13: Comparison of the measured surface parameters to the subjects' ratings of the fifteen real surfaces along the three adjective scales. There is a high degree of correlation between the friction coefficient and the rated slipperiness, and between the acceleration power and the rated roughness. There is weak correlation between the tap response factor and the rated hardness.

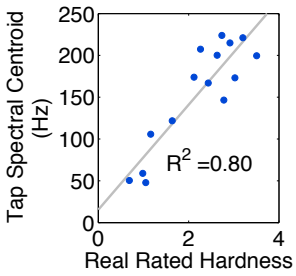

Figure 4.14: Comparison of the tap spectral centroid to the subject's hardness ratings for the fifteen real surfaces. The tap spectral centroid was found to be a better predictor of rated hardness than the tap response factor.

this measurement was found to have a lower than expected correlation with the rated hardness of the real surfaces ($R^2 = 0.22$, $p = 0.076$). Instead, we found the spectral centroid of the tapping transient vibrations had the best correlation with the average rated hardness ($R^2 = 0.80$, $p = 7.30 \times 10^{-6}$), as shown in Fig. 4.14. The average spectral centroid of each surface's tapping transients was calculated across several representative taps as in Eq. (4.5.2). We chose a subset of incoming speeds that were commonly used by subjects when tapping on a virtual surface with the Omni. Tapping transients were calculated at every 10 mm/s in the interval from 50 mm/s

Table 4.3: Correlation of Real Adjective Ratings

Real Adjective	Slipperiness	Hardness	Roughness
Slipperiness	1.00	0.26	−0.91
Hardness	0.26	1.00	−0.05
Roughness	−0.91	−0.05	1.00

Table 4.4: Correlation of Virtual Adjective Ratings

Virtual Adjective	Slipperiness	Hardness	Roughness
Slipperiness	1.00	0.21	−0.74
Hardness	0.24	1.00	0.43
Roughness	−0.74	0.43	1.00

to 100 mm/s; the spectral centroids of the resulting transients were calculated and averaged across all speeds to yield the data plotted in Fig. 4.14.

Table 4.3 shows the correlation of the real adjective ratings, and Table 4.4 shows the correlation of the virtual adjective ratings. There was a high degree of negative correlation between the real ratings of roughness and slipperiness ($p = 3.53\times10^{-6}$) and the virtual ratings of roughness and slipperiness ($p = 0.002$). No other correlations were significant.

Fig. 4.15 shows the relationship between the rated adjectives for the corresponding

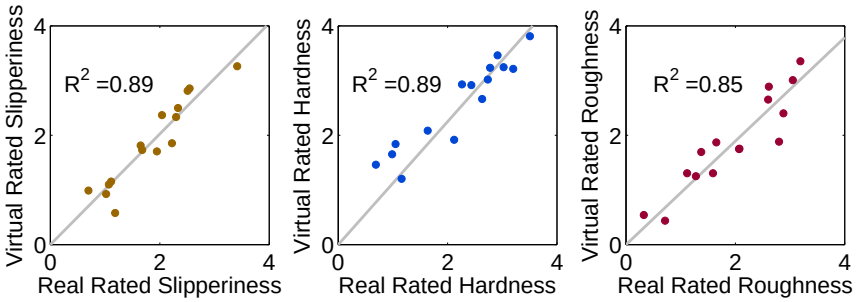

Figure 4.15: Comparison of the subjects' ratings of the control virtual surfaces to their ratings of the fifteen real surfaces along the three adjective scales along the same scale. There is high correlation between the rated characteristics of the real and virtual surfaces for all three perceptual dimensions.

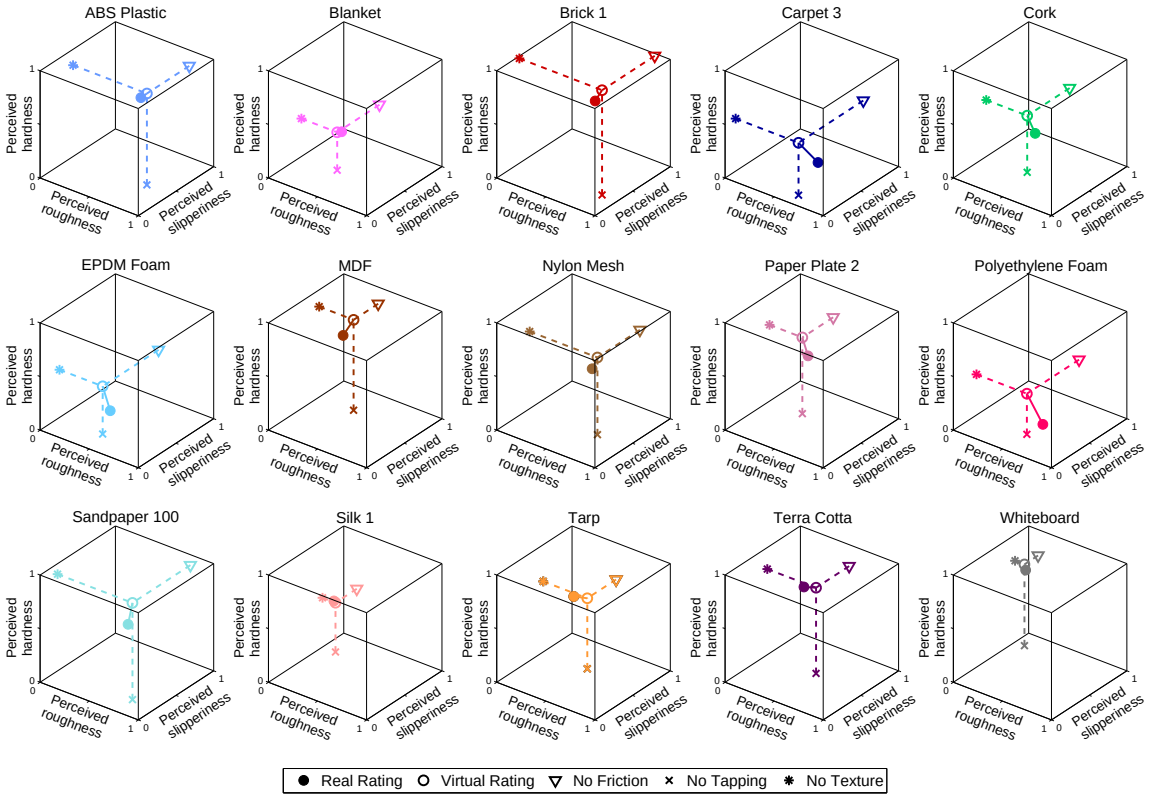

Figure 4.16: Real and virtual surfaces placed in three-dimensional perceptual space. The real and control virtual surfaces are placed using their average adjective ratings. The remaining three virtual surfaces are placed by removing one modeling component (friction, tapping, or texture), which corresponds with the removal of one perceptual component (slipperiness, hardness, or roughness).

real and virtual surfaces. All three sets of adjectives are highly correlated across the real and virtual surfaces including the slipperiness ($R^2 = 0.89$, $p = 1.55 \times 10^{-7}$), the hardness ($R^2 = 0.89$, $p = 1.56 \times 10^{-7}$), and the roughness ($R^2 = 0.85$, $p = 9.78 \times 10^{-7}$).

Fig. 4.16 shows the real and virtual surfaces placed in three-dimensional perceptual space and illustrates the theorized perceptual relationship between the real and virtual surfaces. The real surface and the control virtual surface are placed using the adjective ratings of those surfaces. The distance between these two surfaces is a measure of how similar they feel to one another. The remaining three virtual surfaces are placed

by changing the value of the relevant perceptual component. We place these surfaces under the assumption that the perceived slipperiness is controlled by only the surface friction, the perceived hardness is controlled by only the addition of the tapping transients, and the perceived roughness is controlled only by the texture vibrations. For example, the virtual surfaces without friction are placed by setting the slipperiness to its maximum value; thus they lie in the roughness-hardness plane. Similarly, the virtual surfaces without tapping transients or texture vibrations are placed by setting the hardness or roughness, respectively, to zero.

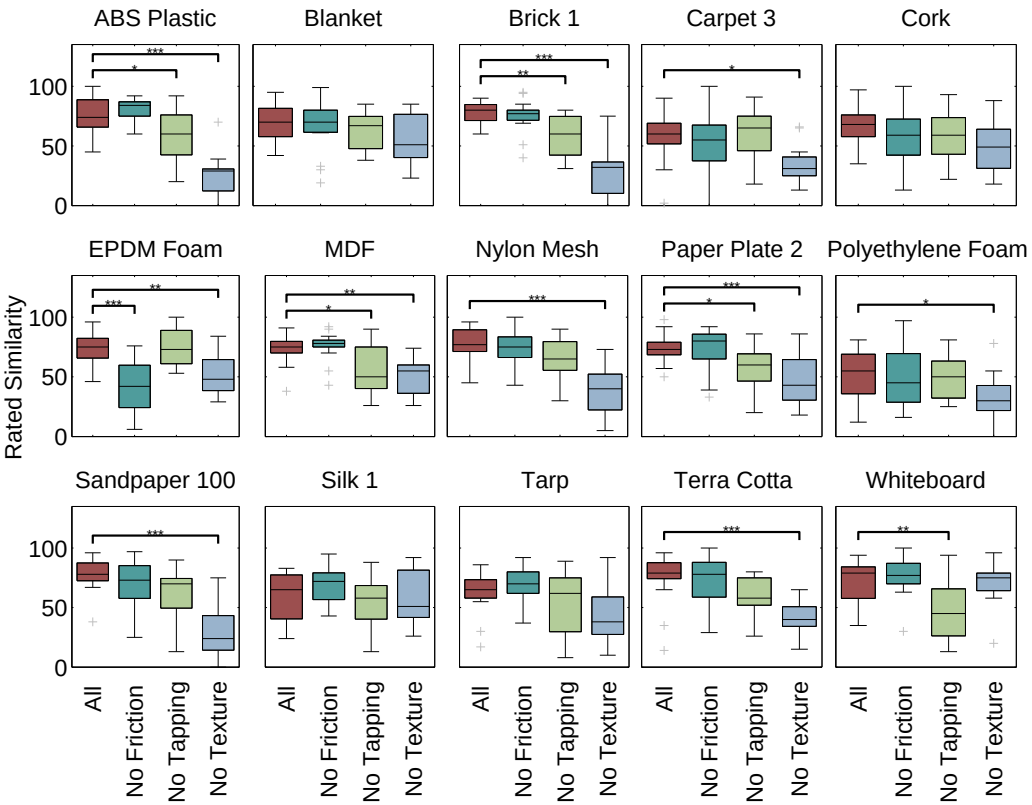

Figure 4.17: Comparison of similarity ratings across the four virtual versions of each surface. Statistically significant differences in similarity from the control virtual surface are marked with a bar and asterisks ($*** \equiv p \leq 0.001$, $** \equiv p \leq 0.01$, $* \equiv p \leq 0.05$).

Similarity Ratings

The similarity ratings are shown in Fig. 4.17. The box plot of the four similarity ratings for each surface provides a visual representation of the relative importance of the three rendering components, which is different for each surface. The similarity ratings between real and virtual surfaces decreased significantly for some conditions in which a rendering component was removed, whereas some other similarities showed no statistically significant difference if a component was removed. Removing the surface friction created a statistically significant decrease in real-virtual similarity for only one surface: EPDM Foam ($p = 1.87 \times 10^{-5}$). Removing the tapping transient response created a statistically significant decrease in similarity for five surfaces: ABS Plastic ($p = 0.048$), Brick 1 ($p = 0.0072$), MDF ($p = 0.019$), Paper Plate 2 ($p = 0.046$), and Whiteboard ($p = 0.0049$). Removing the texture vibrations created a statistically significant decrease in similarity for ten surfaces: ABS Plastic ($p = 4.09 \times 10^{-9}$), Brick 1 ($p = 3.87 \times 10^{-9}$), Carpet 3 ($p = 0.042$), EPDM Foam ($p = 0.004$), MDF ($p = 0.001$), Nylon Mesh ($p = 1.10 \times 10^{-7}$), Paper Plate 2 ($p = 5.83 \times 10^{-4}$), Polyethylene Foam ($p = 0.047$), Sandpaper 100 ($p = 4.91 \times 10^{-8}$), and Terra Cotta ($p = 1.05 \times 10^{-4}$). None of the differences in real-virtual similarity ratings were significant for four surfaces: Blanket, Cork, Silk 1, and Tarp.

The condition that resulted in the maximum average similarity rating was determined for each surface. Fig. 4.18 shows the similarity ratings for each surface for the condition with the highest average rating. Although some of the three conditions in

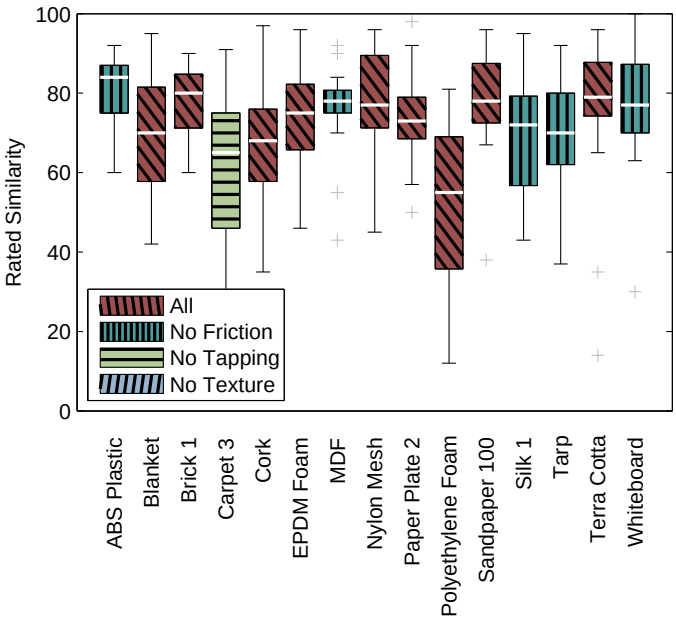

Figure 4.18: Similarity ratings for the condition with the highest average similarity rating calculated separately for each surface. Although some rendering conditions resulted in higher average similarity than the control surface, none of these differences were statistically significant.

which a component was removed have a higher average similarity, the difference is not statistically significant ($p > 0.05$).

For subsequent analyses, each subject's similarity ratings were normalized by the standard deviation of their ratings to compensate for individual differences in subjective rating scales. The normalized similarity ratings were then averaged across all subjects.

To determine how the similarity ratings change when the components are turned off one at a time, we first looked at the change in similarity rating as a function of the surface's measured parameters, as shown in Fig. 4.19. The percent differences were

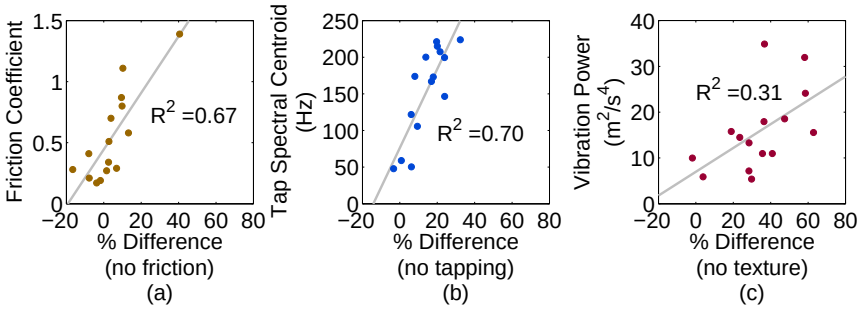

Figure 4.19: Relationship between the surfaces' physical parameters and the percent difference in similarity ratings resulting from the removal of the relevant modeling component.

calculated as follows:

$$\%\text{diff}_x = \frac{S_1 - S_x}{S_1} \tag{4.6.1}$$

where S_1 is the average normalized similarity rating when all model components were displayed and S_x is the average normalized similarity rating when the component of interest was removed. The measured friction coefficient was found to highly correlate to the percent difference in the similarity ratings when the friction was turned off ($R^2 = 0.67$, $p = 1.95 \times 10^{-4}$). The tapping spectral centroid was highly correlated with the percent difference in the similarity ratings when the tapping transients were turned off ($R^2 = 0.70$, $p = 1.12 \times 10^{-4}$). There was a weak correlation between the acceleration power and the percent difference between the similarity ratings when the texture vibrations were turned off ($R^2 = 0.31$, $p = 0.03$).

Lastly we looked at the change in similarity rating as a function of the adjective ratings of the real surfaces, as shown in Fig. 4.20. The percent differences were calculated as in Eq. (4.6.1). There was a weak inverse correlation between the average rated slipperiness and the percent difference between the similarity ratings when the

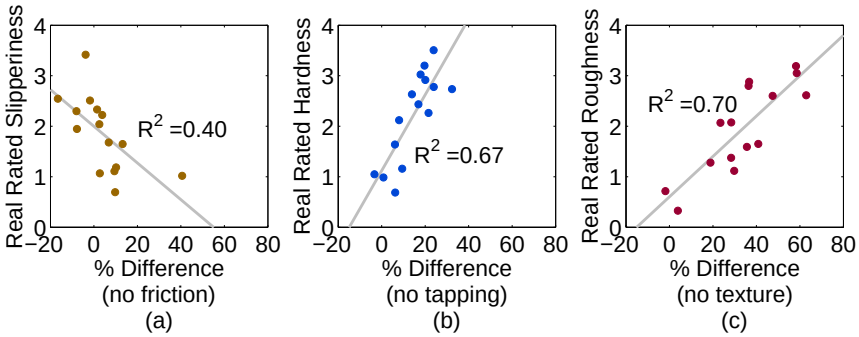

Figure 4.20: Relationship between the average adjective ratings of the real surfaces and the percent difference in similarity ratings resulting from the removal of the relevant modeling component.

friction was turned off ($R^2 = 0.40$, $p = 0.01$). The average rated hardness was highly correlated with the percent difference in the similarity ratings when the tapping transients were turned off ($R^2 = 0.67$, $p = 1.94 \times 10^{-4}$). The average rated roughness was found to highly correlate with the percent difference in the similarity ratings when the texture vibrations were turned off ($R^2 = 0.70$, $p = 9.39 \times 10^{-5}$).

4.6.4 Discussion

By evaluating the change in similarity between real and virtual surfaces induced by the removal of individual modeling components, the study provided important insights into the strengths of our modeling approach and the importance of the perceptual dimensions (slipperiness, hardness, and roughness) to the realism of haptic virtual surfaces. The high correlation between the real and virtual adjective ratings for all three scales (Fig. 4.15) indicates that the modeling and rendering methods presented in this Chapter capture all three perceptual dimensions. This correlation shows a

marked improvement over the rendering methods presented in Chapter 3, which accurately captured only the roughness of the surfaces by modeling and displaying the texture vibrations. Using all three surface model components accurately captures the three perceptual dimensions of surface interactions with a tool.

However, Fig. 4.17 showed that the rendering components are not equally important to the real-virtual similarity of all surfaces. A cursory glance shows that displaying the texture vibrations appears to be the most important because removing texture vibrations creates a statistically significant decrease in similarity for the largest number of surfaces. However, removing the texture vibrations did not result in a significant change in similarity for five surfaces. Furthermore, removing the friction or tapping transients creates a significant decrease in similarity rating for some surfaces, indicating that the importance of the separate modeling components is complex. As suggested by Fig. 4.19, the relative difference in similarity ratings depends on the physical parameters of the surface.

Slipperiness

The percent difference in similarity rating when friction is removed significantly correlates with the measured friction coefficient, as shown in Fig. 4.19a. In simple terms, rendering friction is more important for surfaces with high friction. Interestingly, the similarity ratings for surfaces with the lowest coefficients of friction (Whiteboard: $\mu_k = 0.17$, MDF: $\mu_k = 0.19$, ABS Plastic: $\mu_k = 0.21$, Silk 1: $\mu_k = 0.28$, and Tarp:

$\mu_k = 0.41$) improve when friction is removed. This negative percent difference suggests that the friction of these virtual control surfaces was too high. One reason for this discrepancy is the presence of friction in the motors of the Omni, which was not taken into account when rendering the surfaces. Thus, when the modeled surface friction was added to the Omni's internal friction, the subjects perceived the surface friction as being too high. This finding highlights the need to take the frictional properties of the device into consideration when rendering virtual surfaces.

Although Fig. 4.13 shows that the real ratings of slipperiness are moderately correlated with the physical measurements of surface friction, they show only weak correlation to the percent difference in similarity ratings when the surface friction was removed (Fig. 4.20a). This discrepancy indicates that friction is not the only contributor to the perceived slipperiness of the surface, which violates the assumption we made in creating Fig. 4.16. This observation is supported by the high correlation seen between the real ratings of roughness and slipperiness in Table 4.3 and the high correlation seen between the virtual ratings of roughness and slipperiness in Table 4.4. Thus, the surface roughness, and consequently the texture vibrations, likely play a role in the perceived slipperiness of a surface. This phenomenon was also uncovered by the human subject study results in [26] and has been studied in [13]. Thus, to fully capture the slipperiness perceptual dimension of a surface, both the surface friction and the texture vibrations must be displayed on the virtual surface.

Hardness

Contrary to our expectation that hardness would correlate well with the tap response factor defined in [32], this study found that the spectral centroid of the tapping transient vibrations was the best predictor of perceptual hardness ratings (Fig. 4.14). This relationship results from differences in physical properties of the tool-hand-surface system for hard and soft surfaces. As discussed in [108], impact with high stiffness surfaces elicits wide-bandwidth transient vibrations, and impact with low stiffness surfaces elicits lower-bandwidth vibrations that are damped out quickly. Therefore, the tapping transient vibrations for hard surfaces will have a higher spectral centroid than tapping transient vibrations for soft surfaces. These trends can be observed in the DFTs shown in Fig. 4.3.

The percent difference in similarity rating when the tapping transients are removed is significantly correlated with the measured tapping spectral centroid (Fig. 4.19b) and the rated hardness (Fig. 4.20b). Only one surface had a negative percent difference: Carpet 3. Since this surface had the lowest tap spectral centroid ($SC_{tap} = 47.8$ Hz), it was physically the softest of the 15 tested surfaces. The negative percent difference indicates that virtual Carpet 3 was perceived as too hard, which may have been because the underlying stiffness of the surface was too high. All virtual surfaces were rendered using the same underlying stiffness (0.25 N/mm). This finding highlights the need to take the stiffness of the physical surfaces into account when rendering the virtual surfaces. Displaying tapping transients can make a virtual surface feel

harder, but they are not needed for very soft surfaces that can simply be rendered using Hooke's law.

Roughness

The percent difference in similarity rating when the texture vibrations were removed is only weakly correlated to the measured power of the acceleration signal, as shown in Fig. 4.19c. This disparity indicates that the power was not the only important component of the vibration signal; subjects were probably also paying attention to the signal's spectral properties. However, the percent difference in similarity ratings when the texture vibrations were removed was strongly correlated to the rated roughness of the real surfaces (Fig. 4.20c), which supports the conclusions of previous work that the roughness information is encoded in the vibrations [10, 13, 58, 119]. Furthermore, the real roughness ratings were correlated with the acceleration power (Fig. 4.13), which shows that the power of the vibrations is important in determining roughness. Therefore, to display the roughness of a surface, it is necessary to model and render the texture vibrations. For very smooth surfaces, however, displaying the texture vibrations is not needed to create a realistic virtual surface.

The results of this study show that the importance of rendering the surface friction, tapping transients, or texture vibrations depends on the physical surface being modeled. When deciding whether to include a rendering component or not, researchers should balance the potential increase in realism with the computational and hard-

ware costs of adding that component. To further increase realism, researchers should also take into account the mechanical properties of the rendering hardware, such as friction and bandwidth, when displaying the modeled components to the user.

4.7 Summary

This Chapter presented methods for modeling and rendering haptic virtual surfaces from data recorded during tool-surface interactions. The virtual surfaces were created using a combination of friction, tapping transient, and texture vibration models to capture the full haptic experience. First we presented the data recording and modeling steps for the three component models, followed by a description of the steps necessary to render the virtual surfaces using an Omni force-feedback device augmented with a Haptuator. A Coulomb friction relationship was fit to data recorded from dragging on the surface and was rendered using a stick-slip Dahl friction model. The tapping vibration transients were modeled from data recorded during tapping on the physical surfaces at various speeds. During rendering, the tapping transients are displayed as momentum-cancelling force transients through the motors of the Omni. Texture models were created to represent the vibrations induced in a tool as it is dragged across a textured surface. The recorded vibration signal is modeled as a piecewise autoregressive process, and the individual AR models are labeled with the force and speed at which the data was recorded. During rendering the AR models are used to generate a synthetic texture vibration signal that responds appropriately to the

forces and speeds used to explore the virtual surface. The vibration signal is displayed through a voice-coil actuator attached to the tip of the Omni.

We ran a study to test whether it is necessary to create haptic texture models that respond to both user force and speed. Our analysis showed that removing the speed responsiveness of a texture caused a statistically significant decrease in realism, whereas removing the force responsiveness of a texture did not significantly alter the realism. Therefore, creating haptic textures that respond only to user speed should be sufficient for achieving realism. Our results further showed that, although both force and speed affect the texture signal's power and spectral properties, the changes due to speed are more salient to users than the changes due to force. Researchers can use the results of this study to decide whether to include force and speed responsiveness in their haptic textures.

We ran a second study that evaluated the improvement in similarity between real and virtual surfaces that is created by the addition of friction, tapping, and texture cues. Subjects rated all real and virtual surfaces along three adjective scales (slipperiness, hardness, and roughness), and subjects also rated the similarity between pairs of real and virtual surfaces. Four virtual surfaces were created for each real surface: one control surface in which all model components were displayed and three surfaces in which the model components were turned off one at a time. The study results indicated that the combination of surface friction, tapping transient, and texture vibration models fully captured the real surfaces' slipperiness, hardness, and roughness.

However, the importance of the three model components was not constant across surfaces. We found that the importance of the three separate modeling components had an approximately linear relationship with the physical parameters and the adjective ratings of the real surfaces. Intuitively, rendering a given component of a surface is important when that surface property is strong. Researchers can use the results of the reported study to decide whether to include friction, tapping, and/or texture in their haptic applications.

Chapter 5

Displaying Texture and Friction via Haptic Augmented Reality

The three-dimensional virtual surfaces presented in Chapter 4 accurately captured the physical surfaces' roughness, hardness, and slipperiness. These virtual surfaces were created through a combination of texture vibration, tapping transient, and surface friction models. The choice of a force-feedback haptic device as the rendering hardware allowed us to display normal and lateral forces, which had many benefits over the vibration-only rendering used in the tablet system of Chapter 3. However, as discussed in Section 4.6, the selected force-feedback device had several disadvantages, including limited virtual object stiffness and undesired friction and vibrations. We address these limitations in the novel haptic rendering system presented in this Chapter. As in past Chapters, our goal is to simulate the experience of touching

diverse real surfaces through a tool as realistically as possible.

Due to computational time delays and quantization in position measurements, impedance-type force-feedback devices cannot stably render a virtual object with a stiffness high enough to match that of rigid objects [31]. In the system in Chapter 4, we displayed tapping transients when the user first came into contact with the virtual surface. These transient forces increased the perceived hardness of the surfaces, but the underlying stiffness remained too low. This discrepancy was noticed by subjects when they statically pressed into the virtual surface, which degraded the overall virtual experience. Fully matching the hardness of physical objects is critical to increasing the realism of the interaction. Thus, we have designed a system that uses a stiff physical object in the place of a virtual object. Overlaying haptic texture and friction signals when the user touches the physical object creates a haptic experience that is a mix of real and virtual sensations. Such systems are referred to haptic augmented reality and share many parallels to visual augmented reality systems such as head-mounted displays [47]. Haptic augmented reality systems can include any combination of real or virtual haptics and vision. However, many existing systems center on force-feedback devices (e.g., [4, 8, 14, 40, 47, 49, 51, 105, 117]).

The results from the human subject study in Chapter 4 indicated that undesired friction and vibrations in the force-feedback device further degrade virtual interactions. In that study, the friction of the virtual surfaces was perceived as too high when the modeled friction was displayed on top of the internal friction of the device.

Furthermore, the vibrations of the Omni degraded the feel of the virtual textures. For surfaces with texture vibrations that were low in power (e.g., whiteboard), subjects did not notice a difference between the vibrations they felt when the texture was turned on or off because the texture vibrations were too small to distinguish from the device's mechanical vibrations. These types of issues with undesired friction and vibrations are common to many force-feedback haptic devices, especially those that are cable driven. Rotary motors contribute additional vibrations through cogging torque ripple, which is more prominent in low-cost motors [90]. Therefore, to avoid unwanted friction and vibrations, we must not use cables and rotary motors.

This Chapter presents a novel haptic augmented reality system we have engineered for displaying texture and friction on the surfaces of a three-dimensional physical object. This technical approach deviates significantly from traditional haptic rendering, but the goals are the same: measure how the user's tool is contacting the object, output realistic texture vibrations, and output realistic friction forces. Section 5.1 reviews prior work in the field of haptic augmented reality. Section 5.2 describes our method for determining the tool contact position, speed, normal force, and friction force from a single six-axis force-torque sensor and a three-dimensional geometric model of the object. Section 5.3 describes the design of the tool's critical components for reducing undesired friction and vibrations. Section 5.4 reviews our method for displaying haptic texture vibrations. Section 5.5 describes the mechanism we created for modulating the friction between the tool and the object. Finally, the contributions

of this Chapter will be summarized in Section 5.6.

5.1 Background

The field of visual augmented reality has recently experienced rapid growth, fueled by the growing popularity and technological advances of smart phones and virtual reality headsets [106]. Visual augmented reality overlays visual information, particularly virtual objects, on real-world scenes. This concept of mixed real and virtual environments has been expanded to include haptics as well. Jeon et al. created a taxonomy to describe the possible combinations of real and virtual vision and haptics that comprise the realm of haptic augmented reality [47]. Users can touch a completely real object, a completely virtual object, or a real object with virtual haptic sensations added to augment how it feels. A key challenge lies in creating a scene where the visuals and haptics work seamlessly together.

Many researchers have focused on the challenge of adding virtual objects to a visually real scene. In the majority of these systems, the user feels and interacts with the virtual objects through a force-feedback haptic device. Vallino and Brown created one of the first systems of this type, which allowed users to manipulate virtual objects that were visually overlaid on a physical environment [105]. A Phantom force-feedback device was used to display the shape and weight of the virtual objects. Aleotti et al. created a system for manipulating virtual objects using a Falcon force-feedback device [4]. This system allowed for both virtual and real objects in the

augmented environment, and it handled the real-virtual interactions between objects. Ye et al. created the VisHap system to interact with virtual objects using a Phantom force-feedback device [117]. Unlike many systems that include force-feedback devices, however, the user did not hold the Phantom. Instead, the tip of the Phantom followed the motion of the user and contacted his or her hand only when he or she made contact with a virtual object. This paradigm allowed for more distinct virtual objects because the user felt external sensations only when touching the virtual objects.

Haptic augmented reality systems have also been developed to virtually simulate specific tasks. Bianchi et al. created a haptic augmented reality ping-pong game [14]. This system involved a head-mounted camera to capture the real scene. The virtual object was superimposed on the image, and the augmented scene was then shown using a head-mounted display. Bayart et al. created an augmented reality system for virtual painting using an Omni force-feedback device to control the virtual tool and provide haptic feedback [8]. The Omni was visually hidden, and a virtual tool (i.e., a pencil or a paintbrush) was displayed on the screen. The user then virtually drew or painted on physical objects. Harders et al. created a haptic augmented reality medical simulator using a physical mock-up of a patient's leg [40]. A portion of the physical leg was missing, which was then created virtually through the addition of graphics and haptics. Open surgery was simulated by allowing users to cut into the virtual portion of the leg using a Phantom force-feedback device.

Another area of haptic augmented reality involves changing how a physical object

feels through the addition of haptic feedback. For example, researchers have created systems that change the perceived friction between a tool and the physical surface. Jeon et al. modulated the friction of a physical surface using a Phantom force-feedback device [51]. The Phantom device included a force sensor attached at the tip, which measured the friction as the user dragged the tip across a physical surface. The Phantom created an additional lateral force to reach a target friction force. In another approach, Wintergerst et al. created a set of styli that could actively alter the friction between the stylus tip and a touch screen [114]. An electromagnetic coil was used to attract a rolling ball to the body of the stylus, which would increase the apparent surface friction. This friction control system was used to provide feedback during dragging gestures, including guiding the user, simulating virtual mass, and enhancing drawing by simulating different tools.

Researchers have also created haptic augmented reality systems to change the perceived hardness of physical objects. Jeon and Choi modulated the stiffness of a physical object using a Phantom force-feedback device [47]. A force sensor was attached to the tip of the Phantom to measure the user's applied force. The Phantom was then used to generate additional force to increase the object's perceived stiffness. This stiffness augmentation was extended to allow for three-dimensional interactions with physical objects [48, 49], and it was further extended to allow the perceived stiffness to be changed while squeezing a physical object using two Phantom devices [50]. Hachisu et al. altered the perceived hardness of a physical surface by displaying tap-

ping transients using a voicecoil actuator [37]. Innate vibrations from tapping on the physical surface were removed by using an elastic sheet so that the physical hardness was affected only by the tapping transients.

Haptic augmented reality systems have also been created to change the perceived texture of physical objects. Bau and Poupyrev created the REVEL system to change the perceived texture of objects touched with a bare finger [7]. Texture was added using reverse electrovibration, which involves injecting a weak AC electrical signal into the user's body, creating an oscillating electrical field around the user's fingers. The user then touches physical objects that have been coated with an insulator-covered electrode. The properties of the AC signal determine the surface texture that is felt. Yoshimoto et al. altered perceived texture by displaying electrical stimuli through electrodes attached to the user's finger [118]. Asano et al. changed the perceived roughness of textured surfaces by playing vibrations through a voicecoil actuator attached to the user's finger [5]. The perceived roughness could be increased or decreased depending on the properties of the vibrations that are played.

Past work in haptic augmented reality has tended to focus on changing the perception of only one haptic attribute. This Chapter presents a novel haptic augmented reality system that alters both the perceived texture and friction of a physical object.

5.2 Interaction Condition Measurements

This Section describes the system we have designed to measure the user's interaction conditions during rendering. Our previous haptic surface rendering schemes that were discussed in Chapters 3 and 4 required measurements of the user's normal force and scanning speed to generate the haptic texture and friction signals. To utilize the same texture and friction models, this new rendering system also requires the ability to measure these interaction conditions in real time. This Section describes the hardware and methods required to collect this data.

5.2.1 Hardware Description

When determining the best hardware for measuring the user's interaction conditions, we had to take into account both the measurements required and the overall user experience. Displaying both texture and friction on the surfaces of a physical object requires measurements of the user's normal force and scanning speed. Therefore, we needed hardware that was capable of measuring three axes of contact force data and at least two axes of position data. While we needed to include all of these measurements, however, we also wanted to create a device that was not overly large or cumbersome to use. Our haptic recording tool discussed in Section 2.2.1 was capable of measuring the signals of interest, but it was not suitable for our augmented reality rendering system because it is bulky and does not have space for friction actuation, due to the placement of the force sensor.

Figure 5.1: The force/torque sensor system used to determine interaction conditions. (a) The sensor's axes and origin. (b) An acrylic cube is rigidly attached to the sensor to serve as the physical object to be augmented with texture and friction.

Instead of placing the haptic interaction sensors inside the tool, we decided to offload the sensing by placing a six-axis force/torque transducer (Nano17 SI-25-0.25, ATI Industrial Automation Inc.) underneath the physical object. The force sensor and its axes are shown in Fig. 5.1. For this prototype, we used a 25.4 mm cube as the physical object, rigidly mounted to the force sensor via a custom 3D-printed adapter, as shown in Fig. 5.1(b). Other objects could be used instead. The force applied by the user to the object is measured by the Nano17. This sensor has a force resolution of 1/160 N and torque resolution of 1/32 N·mm along all axes. The signals are read by the computer at 1000 Hz through a 16-bit USB analog-to-digital DAQ device from National Instruments. The sensor is zeroed with the object attached, and the voltages are low-pass filtered at 20 Hz to remove noise.

The three-axis force measured by this sensor will allow us to calculate the normal force and friction force of the interaction, as discussed in Section 5.2.4. But how can we measure the point of contact? As discussed in Section 5.2.3, we can use the measured torque signal to estimate the point of contact. Consequently, the size of the physical object that can be used is limited by the sensor's maximum rated torque, which is ±250 N·mm about each axis. As discussed in Section 4.5.4, the average force applied by a user when dragging a tool across a textured surface was 1.1 N [25]. The same study data showed that the maximum force applied by any of the users was 4.3 N. Therefore, it is reasonable for us to assume that the user of our system will not apply a force higher than 5 N. If we assume the object is centered in the x- and y-directions shown in Fig. 5.1(a) and does not extend below the top of the sensor, the maximum size of the object is $10\,\text{cm} \times 10\,\text{cm} \times 5\,\text{cm}$.

5.2.2 Force Threshold

During rendering, we measure the three axes of force and torque information as the user drags a tool along any portion of the physical object. It is important to correctly determine whether the user is touching the object because the realism would be significantly degraded if texture vibrations were displayed when the tooltip was in free space. Contact can be detected by comparing the magnitude of the measured force vector to a force threshold value. This force threshold was chosen by clamping the force sensor to a table and monitoring the measured force as noise was added

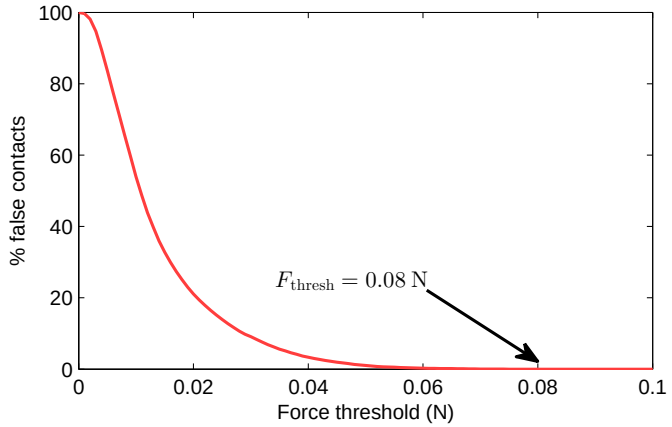

Figure 5.2: Percentage of false contacts as a function of the force threshold level.

to the force signals by tapping on the table near the sensor. This noise was added to simulate the system's real working environment where vibrations could be present due to someone walking by or bumping into the table. Fig. 5.2 shows the percentage of false positive contacts as a function of force threshold values. We selected the force threshold to be $F_{\text{thresh}} = 0.08$ N as this was the minimum value for which there were no false contacts.

5.2.3 Position Measurements

The haptic recording system described in Section 2.2.1 uses a magnetic motion tracking sensor to measure the tool's speed and position. However, there are many limitations and disadvantages to using this type of sensor, including its low sampling rate, sensitivity to ferromagnetic material, and sensitivity to magnetic fields created by electromagnetic actuators. To avoid these limitations in our augmented reality system, we devised a way to sense tool contact position without requiring an additional

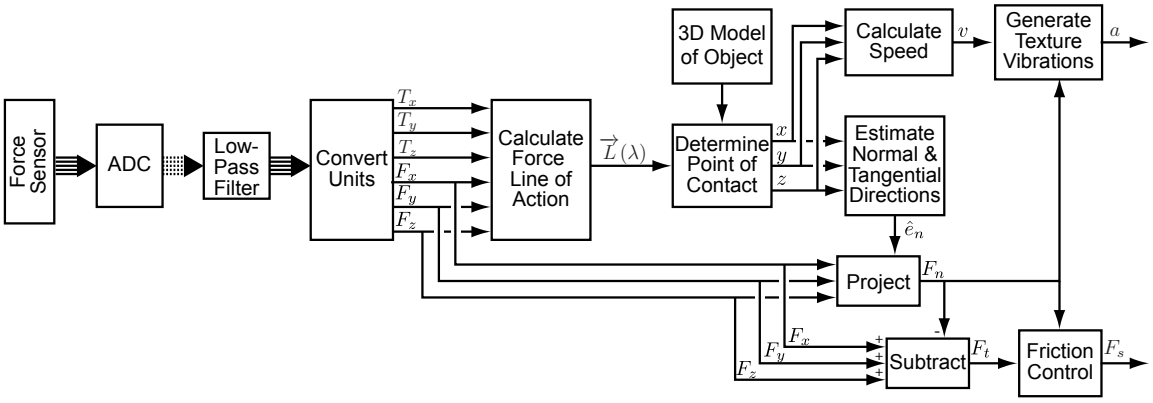

Figure 5.3: A diagram of our system for determining the user's interaction conditions on the augmented reality system. The user's contact position is determined from a three-dimensional geometric model of the object and measurements of the applied force and torque.

sensor. We combine a geometric model of the object with measurements of the force and torque applied by the tool to estimate the tool's contact location. This method for determining contact position through intrinsic contact sensing was first proposed by Bicchi et al. [15] and has been applied to a medical simulator by Riener et al. [94]. Fig. 5.3 displays a flow diagram showing the full process.

A diagram illustrating our method for determining the contact position is shown in Fig. 5.4. The user's applied force is transmitted directly through the rigid object so that the measured force is equal to the applied force. The relationship between the moment $\vec{M}$ and the force $\vec{F}$ measurements from the force/torque sensor is:

$$\vec{M} = \vec{r} \times \vec{F} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \times \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{5.2.1}$$

where $\vec{r}$ is the unknown position vector from the origin of the force/torque sensor

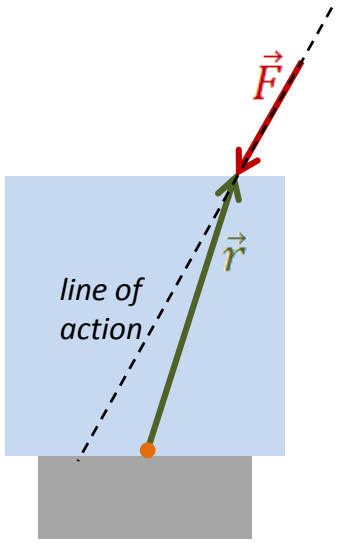

Figure 5.4: Diagram illustrating how contact position is estimated from force and torque measurements.

to the point of contact. This equation cannot be solved directly for $\vec{r}$ because the skew-symmetric matrix representing the cross product is not invertible. However, we can represent the line of action of $\vec{F}$ as the straight line:

$$\vec{L}(\lambda) = \vec{r^*} + \lambda\vec{F} \tag{5.2.2}$$

where λ is a scalar parameter with the units mm/N and $\vec{r^*}$ is any vector from the sensor origin to the line of action. As given in [94], this vector can be calculated as:

$$\vec{r^*} = \frac{\vec{F} \times \vec{M}}{\left|\vec{F}\right|^2} \tag{5.2.3}$$

The point of contact can then be calculated by finding the intersection point(s) between the force's line of action and the surface of a three-dimensional geometric model of the object. Our prototype employs a cube, for which it is trivial to calculate the

intersection points on the faces of the cube. We can represent more complex objects as a mesh and calculate the intersection points between the force line of action and the individual triangles of the mesh. If the line of action passes through more than one point on the surface of the object, the point where the force is in compression is chosen as the point of contact. To avoid multiple potential points of contact, the physical object must be convex. We calculate the tool's speed from a history of the points of contact, which allows us to track the motions made when the user is interacting with the physical object.

Because an additional position sensor is not required, we can determine the contact position of any tool that touches the object. The only requirement is that the tool makes a point contact with the object. This condition is necessary because the tool cannot apply any local moments for the equations to remain valid, nor can it contact the surface in two locations.

5.2.4 Force Measurements

Once the point of contact is determined, the measured force vector can be separated into its normal and tangential components. First, the vector normal to the surface at the point of contact ($\hat{e}_n$) is calculated. The inner product is calculated between ($\hat{e}_n$) and the force vector $\vec{F}$ to find the magnitude of the normal force, F_n. The tangential force vector is calculated as $\vec{F}_t = \vec{F} - \vec{F}_n$. The friction force, F_t, is the magnitude of the force tangential to the direction of motion.

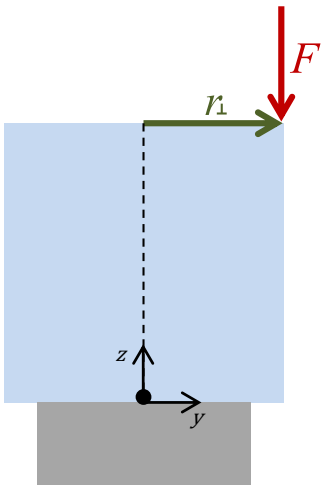

Figure 5.5: Free body diagram showing the two-dimensional case used to determine position repeatability.

5.2.5 Analysis of Position Measurements

The method described in Section 5.2.3 enables us to calculate the tool contact position from measurements of the force and torque vectors acting on the object. To test the limits of our system and determine the efficacy of this position sensing approach, we characterize the accuracy and resolution of the resulting position values. This Section details our theoretical and experimental analysis of the system's position measurements.

Theoretical Position Repeatability

The theoretical position repeatability of our system can be understood using an analysis of the propagation of uncertainty for a simple two-dimensional case. For this analysis we assume a vertical normal force is applied in the negative two-direction ($\vec{F} = -F\hat{\mathrm{k}}$) at a horizontal distance $\vec{r} = r\hat{\mathrm{j}}$ from the origin, as shown in Fig. 5.5. The

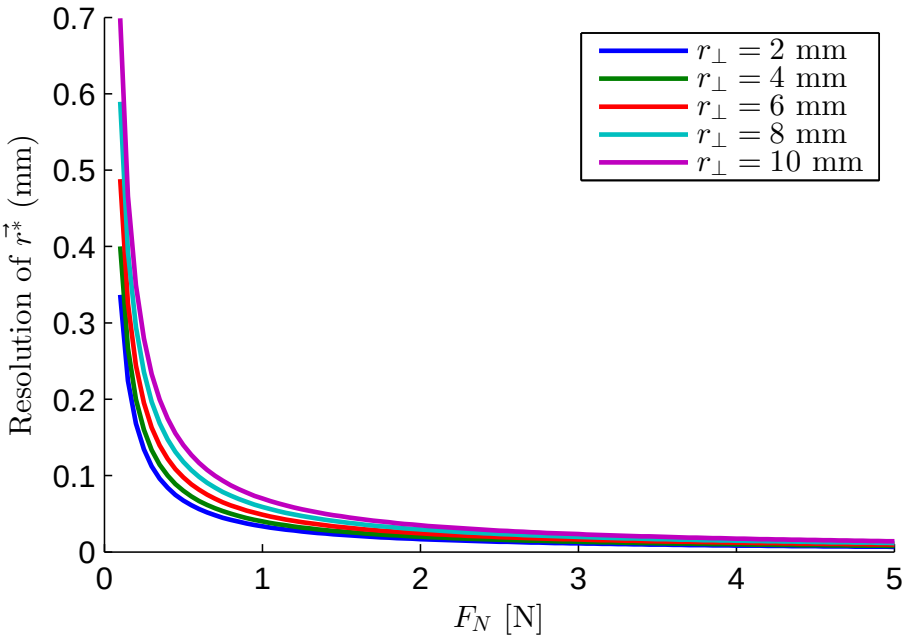

Figure 5.6: Theoretical position repeatability estimated using error analysis of a simple one-dimensional case of the contact position calculations.

resulting moment is $\vec{M} = -M\hat{\mathrm{i}}$. From Eq. (5.2.3), the vector from the origin to the line of action of the force is $\vec{r^*} = \frac{M}{F}\hat{\mathrm{j}}$, which can also be seen by inspection. Using error analysis, the uncertainty in $\vec{r^*}$ is:

$$\Delta\vec{r^*} = \frac{M}{F}\sqrt{\left(\frac{\Delta M}{M}\right)^2 + \left(\frac{\Delta F}{F}\right)^2}\hat{j} \tag{5.2.4}$$

where ΔM is the uncertainty in the sensor's moment measurement and ΔF is the uncertainty in the sensor's force measurement. Therefore, the uncertainty introduced in the calculation for the position depends on the applied force and moment values. Fig. 5.6 shows the theoretical position repeatability for the simple two-dimensional case for different values of $r_\perp$. The plot shows that the uncertainty is significantly higher for lower forces, but is approximately constant for higher forces.

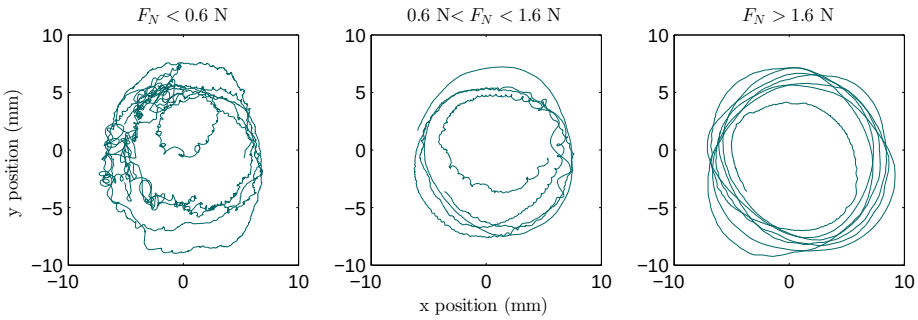

Figure 5.7: Estimated tool position at different normal force levels. The amount of noise in the position decreases as force is increased.

Effect of Force on Position Measurements

This force dependency of the calculated position is evident in the experimental system. Fig. 5.7 shows calculated tool trajectories for the same type of tool motion conducted at three force levels. As predicted above, there is more noise in the position signal when the normal force is low. The estimated trajectories are smoother at medium or high force values.

Static Position Resolution

After determining that the position repeatability is highly dependent on the applied force, we wanted to investigate the system's static position resolution for a constant force. The system's position resolution was measured by recording the contact position as a probe was statically held at fixed points on the top surface of the cube. Five trials were recorded for each of five points on the cube; the points were located at the center and near the four corners of the cube's top surface. For each trial we calculated the deviation of the position measurements from the average measurement for that

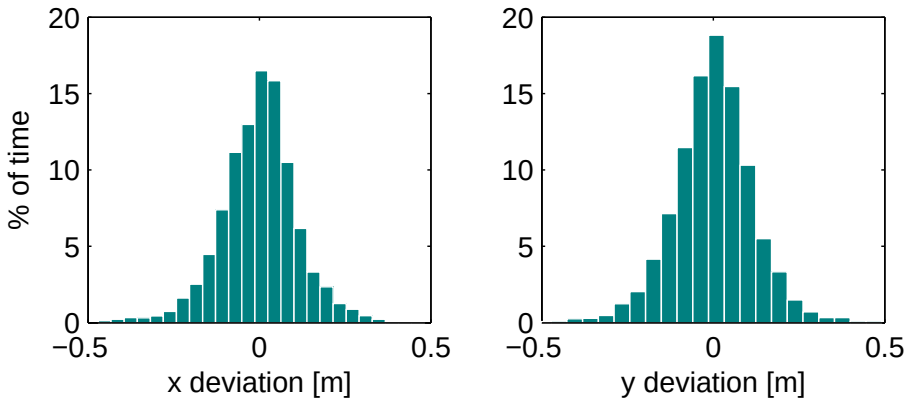

Figure 5.8: Distribution of position errors for static position measurements. The distributions are the same in the x and y directions, indicating that position resolution is the same in all directions.

trial in the x and y directions separately. The first and last 50 ms of each trial was excluded to eliminate any effects of loading and unloading the force sensor. The position errors were combined across all trials and all points. The resulting distribution of position errors is shown in Fig. 5.8. The error did not exceed ± 0.5 mm in either direction for any of the trials. A two-sample Kolmogorov-Smirnov test indicated that the two error distributions were not statistically different between the x and y directions ($p = 0.29$). Furthermore, the error distributions were not statistically different across the different trials or points on the cube. These results indicate that the static position measuring resolution of our system is approximately 0.5 mm, which is the same as the position resolution of the magnetic tracking sensor used in our haptic recording device.

5.3 Pen Design

Although our force and position measurement system works with any tool that has a single point of contact, we wanted to build a custom tool that could give us the cleanest haptic experience with low inherent friction and vibrations. The design of the pen is also crucial to the texture and friction output that will be discussed in Sections 5.4 and 5.5. This Section describes the design and analysis of the resulting pen.

5.3.1 Design Considerations

When rendering a virtual haptic interaction, it is important to start with a smooth base interaction to create free space that feels free [76]. During rendering the user ideally feels only the haptic signals the system is actively displaying, which should not be contaminated with underlying sensations from the haptic device. This contamination was an issue in the Omni-based system studied in Section 4.6, where the device's friction caused the perceived surface slipperiness to be too low when the modeled surface friction was displayed. To avoid similar problems with this haptic augmented reality system, we sought a design with low friction between the pen and the physical object.

Another important design consideration is the level of vibrations felt when dragging the tool across the object's surface. Because we will be changing the perceived texture of the object by playing texture vibrations during dragging, we want the vi-

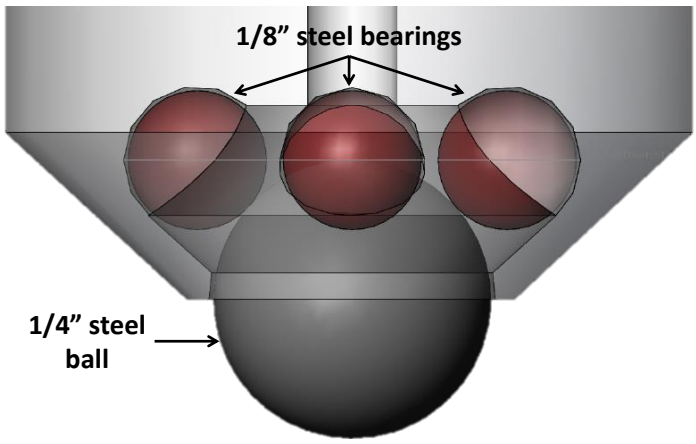

Figure 5.9: Tooltip socket design with four bearings.

brations caused by dragging the pen across the base material to be as low as possible. If these vibrations are too high, they can contaminate the desired texture vibration signal, degrading the overall interaction.

5.3.2 Tooltip Design

To achieve low friction between the pen and any base surface, we designed a pen whose tip contained a rolling ball. The ball must be free to roll smoothly in all directions; therefore, we designed a low-friction socket to contain the ball. Instead of building a single socket out of a low-friction material, we created a system of four metal bearings to allow the ball to rotate, as shown in Fig. 5.9. A single socket would be more susceptible to wear and would have more friction overall than a design that includes bearings. In our socket design, the four bearings share the load of the normal force applied when dragging the tool across a surface.

The four-bearing socket has many benefits over other bearing designs that were

considered. Other proposed designs included lining the entire socket with small bearings. This design did not work well because the bearings rubbed against one another and occasionally jammed. The design shown in Fig. 5.9, however, keeps the bearings separate in their own sockets, so they do not rub together. The resulting socket allows the tooltip to move easily along a surface.

5.3.3 Material Choice

It was essential that the bearings and the central ball roll smoothly with low friction and add minimal vibrations. To roll smoothly, the materials must be rigid and have low coefficients of friction at all interfaces. The materials must also have good abrasion resistance to limit wear so that the system will continue to function over an extended period of time. Delrin and PTFE (Teflon) were considered for the bearings and the center ball because of their low friction. However, the surface finish on the plastic balls was not smooth, causing noticeable vibrations during rolling. Instead we chose chrome steel as the material of the bearings and the center ball because of its surface quality, hardness, and wear resistance. The socket was printed using an Stratasys Objet30 3D printer because of its high resolution (28 μm), high accuracy (0.1 mm), and low friction.

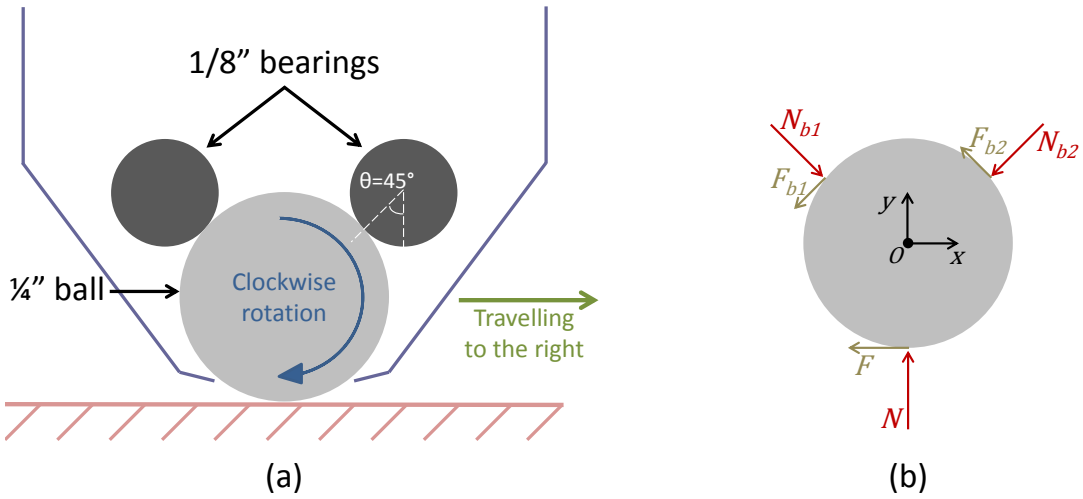

Figure 5.10: (a) Simplified socket design used in analysis. (b) Free body diagram of forces applied to the rolling ball.

5.3.4 Dynamic Analysis

To analyze the dynamic behavior of the socket design, we look at a simplified two-dimensional version of the socket, as shown in Fig. 5.10(a). The ball is rotating clockwise, resulting in horizontal motion to the right. The bearings are fixed in their sockets. The center ball is rolling without slipping on the ground, and it is slipping along the surface of the bearings. Fig. 5.10(b) shows a free body diagram of the forces acting on the system.

Assuming the system is moving at constant velocity, the force and moment balance equations are:

$$\sum F_x = -F - F_{b1}\sin(45^\circ) + N_{b1}\sin(45^\circ) - F_{b2}\sin(45^\circ) - N_{b2}\sin(45^\circ) = 0 \quad (5.3.1)$$

$$\sum F_y = N - F_{b1}\cos(45^\circ) - N_{b1}\cos(45^\circ) + F_{b2}\cos(45^\circ) - N_{b2}\cos(45^\circ) = 0 \quad (5.3.2)$$

$$\sum M_{z_{@o}} = -FR + F_{b1}R + F_{b2}R = 0 \tag{5.3.3}$$

where R is the radius of the ball.

We assume the friction between the bearings and center ball obeys a Coulomb relationship, where the friction force is proportional to the normal force. Therefore, the friction forces can be written as:

$$\begin{aligned} F_{b1} &= \mu_b N_{b1} \\ F_{b2} &= \mu_b N_{b2} \end{aligned} \tag{5.3.4}$$

where μ_b is the coefficient of kinetic friction between the bearings and the ball. The center ball rolls without slipping on the surface.

Because the ball must roll without slip along the surface, the following constraint must also be obeyed:

$$F \leq \mu_s N \tag{5.3.5}$$

where μ_s is the coefficient of static friction between the ball and surface. Although the ball is rolling along the surface, we want the user to feels as though the tip of the pen is sliding over the physical object with a specified coefficient of kinetic friction, μ, to simulate the feel of the selected surface. Thus, we need to ensure that the static friction force F obeys the Coulomb relationship with the associated normal force N, as follows:

$$F = \mu N \tag{5.3.6}$$

where μ is the effective coefficient of kinetic friction between the tooltip and the

surface.

5.3.5 Analysis of the Socket Design

This Section details our analysis of the completed tooltip design to test how well it conforms to our design criteria of low friction and low vibrations during rolling.

Friction in the Socket

The displayed kinetic friction between the tool and a rigid surface should not depend on the surface. Rather, as long as the ball is allowed to roll freely in the socket, the measured surface friction should depend only on the friction between the bearings and the center ball. However, the static friction between the ball and the surface must be sufficiently high for rolling to occur; otherwise, the ball simply slides along the surface. This rolling behavior is an important part of the mechanism for modulating the perceived friction between tool and surface, as will be discussed in Section 5.5.

Soft surfaces will have a higher effective kinetic friction than very rigid surfaces. This disparity occurs because softer surfaces deform slightly during rolling. Extra force is needed to overcome the rolling resistance that results from the deformation of the surface in front of the ball. Therefore, our augmented reality system uses a rigid physical object to ensure smooth, low-friction interactions.

To test the level of friction in the socket, the pen was dragged across a smooth surface mounted to the Nano17 force sensor, and the resulting normal and friction

forces were measured. The effective coefficient of kinetic friction between the ball and the surface was found to be $\mu = 0.17$ by fitting a line to the measurements of normal and friction force. Solving Eq. (5.3.1)–(5.3.3), we find that $\mu_b = 0.12$. This finding sets the lowest coefficient of friction that the haptic augmented reality system can display. This coefficient of friction ($\mu = 0.17$) is lower than the friction of 97 of the 100 modeled textures available in our texture toolkit described in Appendix A. Therefore, the base friction is low enough to allow us to match the friction of almost all of the modeled textures.

Vibrations in the Socket

The physical contact between tool and surface generates unwanted tool vibrations, which affect the feel of the interaction. Therefore, the choice of the underlying object surface, as well as the socket design, was important to the overall system performance. The surface needs to be smooth to minimize the unwanted vibrations. As discussed in Section 5.3.5, the surface also needs to be rigid and have a high enough coefficient of static friction to initiate rolling of the ball. To conform to all of these constraints, we chose a rigid acrylic cube that is covered in a thin layer of rubber coating (Plasti Dip, Performix).

To test the level of inherent vibrations in the system, the tool was dragged over this smooth, rubber-coated surface. The three axes of acceleration were mapped onto a single axis using the DFT321 algorithm [66], and they were high-pass filtered at

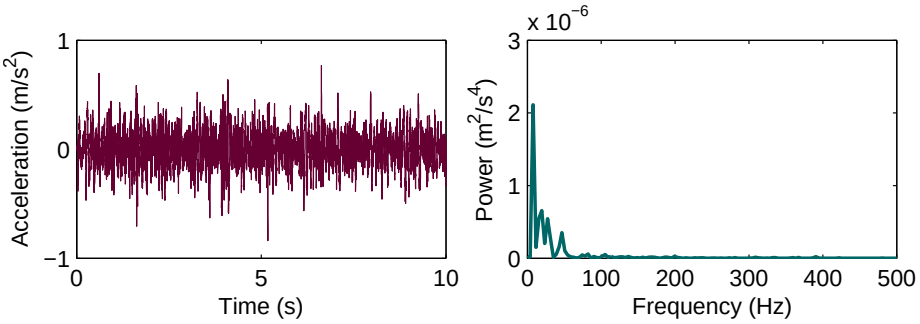

Figure 5.11: Vibrations produced as the tool is dragged over a smooth surface.

20 Hz to remove effects of tool motion. Fig. 5.11 shows the recorded vibrations and the signal's DFT. The vibrations are lower in power than the vibrations for any of the 100 modeled textures available in our texture toolkit described in Appendix A. Although the inherent vibrations are perceptible when no textures are being displayed, they will not mask the virtual texture vibrations themselves. They may cause some degradation in the realism, however, which we expect to be more prominent for textures that are lower in power.

5.4 Augmented Textures

Now that we can measure interaction conditions and present a low-friction, low-vibration base contact experience, we turn our attention to adding textures. The three-dimensional virtual haptic rendering system discussed in Chapter 4 showed that it is simple to map our haptic textures to an arbitrary three-dimensional shape. Rather than needing the exact position of the tool, the texture vibrations depend only on the user's current tangential speed and normal force. This Section describes

our method for adding our haptic textures to this augmented reality system.

5.4.1 Generating Textures

The user's current tangential speed and normal force can be measured using the force-torque sensor placed underneath the object, as discussed in Section 5.2. These values are calculated at a rate of 1000 Hz and are used to determine the appropriate texture model through interpolation as described in Section 2.4.1. The next texture output value is calculated at every timestep as described in Section 2.4.2.

5.4.2 Hardware

After the texture signal is generated, it is scaled and converted to a voltage, which is output through an analog output pin on a Sensoray 626 PCI card at a rate of 1000 Hz. The output voltage is then passed through a linear current amplifier with a gain of 1 A/V.

This signal is then used to drive a voicecoil actuator (Haptuator Redesign, Tactile Labs TL-002-14R) that is rigidly attached to the pen, as shown in Fig. 5.14. The actuator was attached perpendicular to the axis of the pen to maximize the propagation of the vibrations through the tool and to the user's fingertips. If the actuator was aligned with the long axis of the pen, a significant portion of the vibrations would be damped out because of the contact between the tooltip and the rigid object.

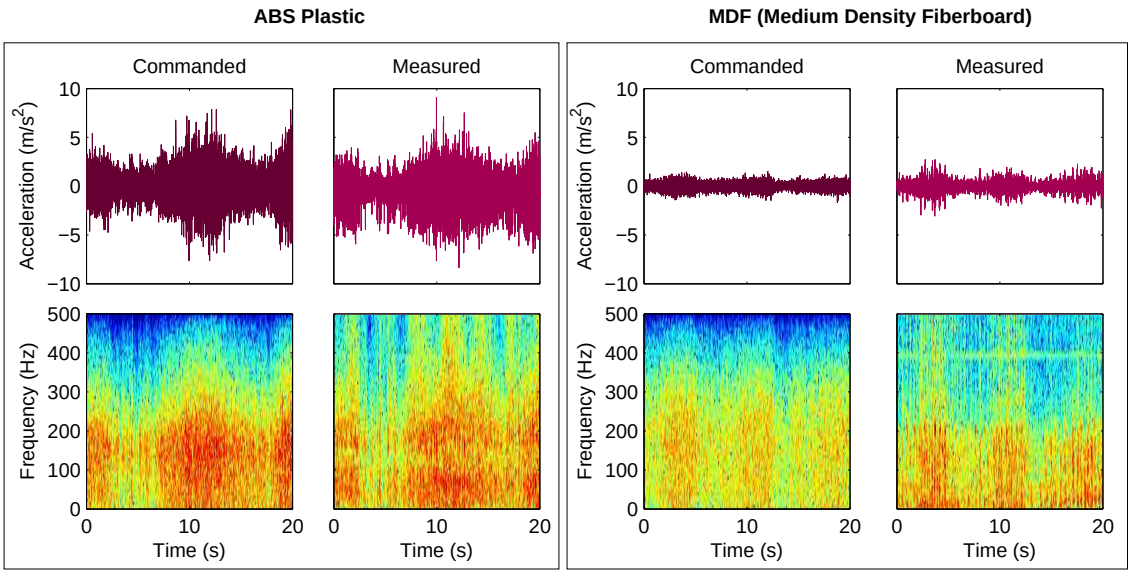

Figure 5.12: Comparison of commanded and measured accelerations for two materials.

5.4.3 Analysis

This Section details our analysis of the texture rendering part of the haptic augmented reality system. We test how closely the output vibrations match the commanded vibrations. Then we analyze how playing the texture vibrations affects measurements of tool position.

Commanded vs. Measured Vibrations

To test the texture vibration output of the system, we compare the commanded texture vibration signal with the measured vibrations felt by the pen. Fig. 5.12 shows the comparison for two materials, ABS Plastic and MDF, which were chosen as representative rough and smooth textures, respectively. The vibrations were synthesized and recorded in real time using measurements of the user's normal force and scanning

speed as the experimenter dragged the pen across the smooth base material. A range of forces and speeds were used in order to compare the vibrations for different interaction conditions. Three axes of acceleration were recorded using two ±18 g ADXL321 accelerometers, and they were mapped onto a single axis using the DFT321 algorithm. For each material, the amplitude of the spectrograms was normalized using the maximum power of the commanded and measured signals, so the spectrograms are shown using the same amplitude scale.

The measured vibrations for both ABS Plastic and MDF closely match the commanded vibrations in both power and frequency content, as can be seen in the spectrograms. However, the measured vibrations have higher power at frequencies around 500 Hz for both ABS Plastic and MDF. This disparity in high frequency content is due to aliasing in the recorded acceleration signals, and is not present in the vibrations being played to the user.

Effect on Position and Force Measurements

Displaying the texture vibrations through the voicecoil actuator attached to the pen creates vibrations that propagate through the entire body of the pen. These vibrations will also propagate to the object, which may add high-frequency noise to the force and torque measurements. Since the tool's position is calculated using the force and torque measurements, the vibrations might affect the tool's estimated position as well. Therefore, it was important for us to test what effect, if any, the vibrations have

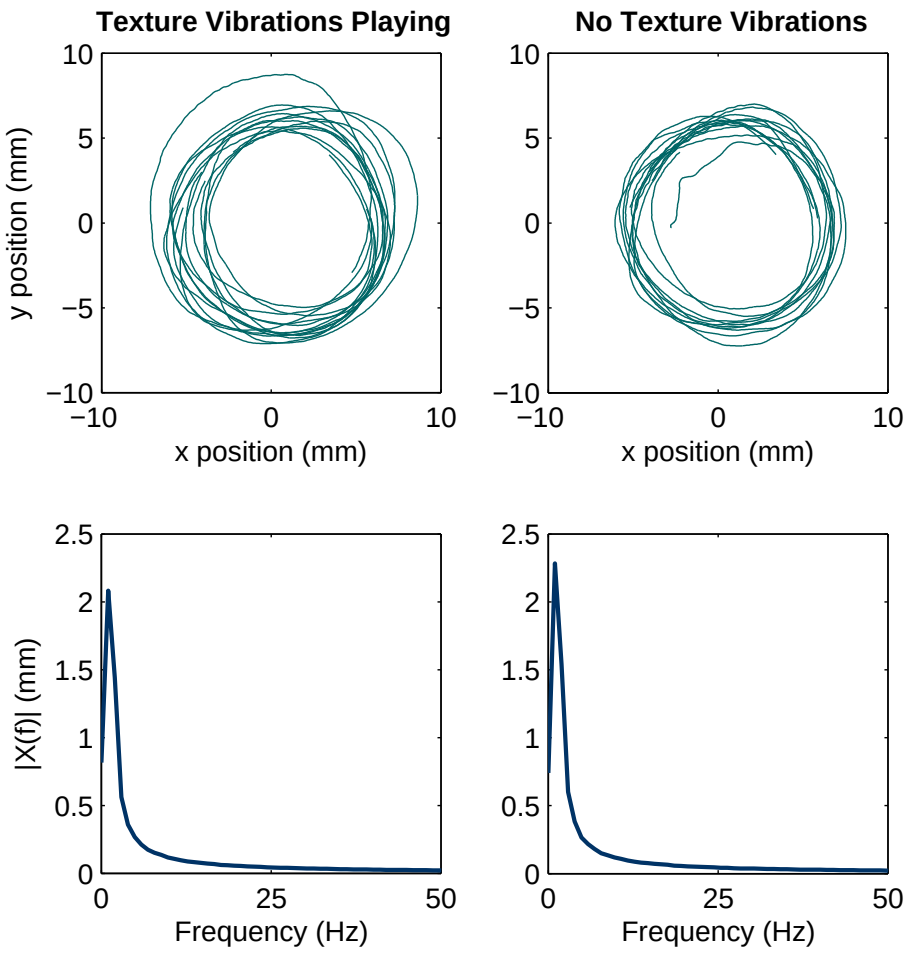

Figure 5.13: Calculated tooltip trajectories when vibrations were playing and when they were turned off. The DFT of the two trajectories indicates that the calculated tool position is not affected by the presence of external vibrations.

on position.

An experimenter dragged the tool across the surface in a circular motion, intermittently turning the texture vibrations on and off. We chose to play the texture vibrations with the highest power (Brick 1) to test the worst case scenario. Fig. 5.13 shows the calculated tooltip position separately when the textures were on and off. The measured tool trajectories are smooth both with and without vibrations playing.

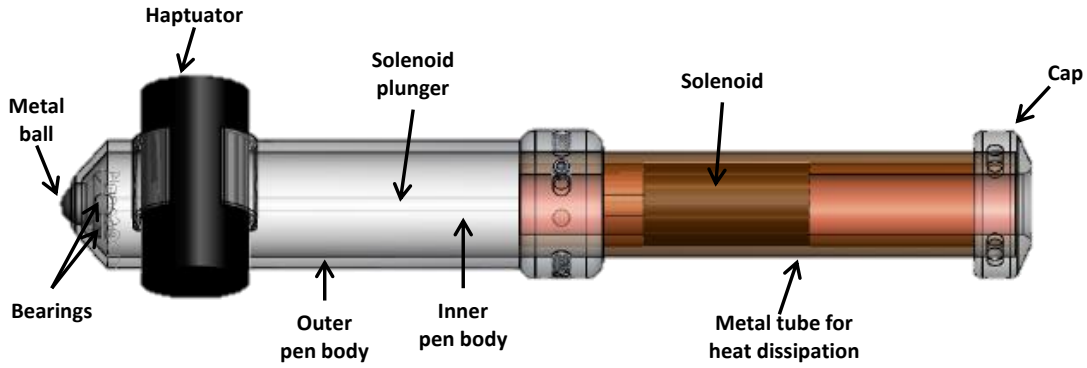

Figure 5.14: Design of haptic augmented reality pen. It can alter the perceived texture of a physical object by displaying texture vibrations through the Haptuator. The friction between the pen tip and object is modulated using a braking force applied to the metal ball by the solenoid.

The discrete Fourier transform (DFT) of the position measurements shows that the frequency content of the trajectories was the same for the two conditions. The peak of the DFT for both condition occurs at about 1 Hz, which was the frequency of the tool's circular motion. These results show that although the texture vibrations can introduce some noise into the force and torque signals, low-pass filtering the signals from the sensor is sufficient to mitigate any effect of this noise.

5.5 Augmented Friction

The study reported in Section 3.3 showed that displaying texture vibrations accurately captures the roughness of the surfaces but does not capture any of the surface's slipperiness. Therefore, to be able to display more than roughness with our augmented reality system, we must include friction. This Section describes the modifications that were made to our pen to allow for the modulation of friction between the tool and the

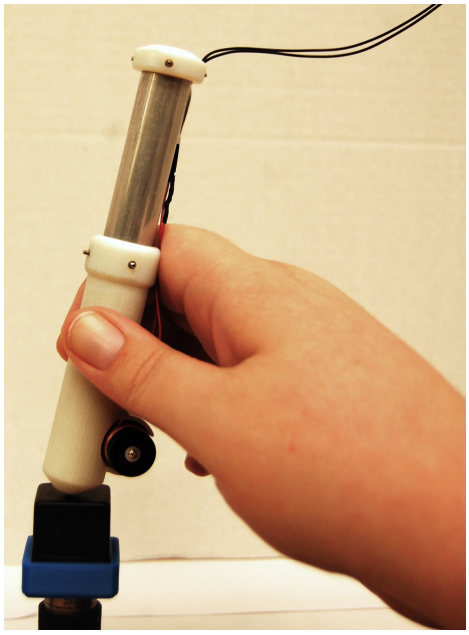

Figure 5.15: Full haptic augmented reality system. An object is rigidly attached to the force sensor, which provides measurements of the tool's contact position, normal force, and tangential force. The pen overlays haptic texture and friction signals on the physical object based on the measured interaction conditions.

physical surface. The design of the pen is shown in Fig. 5.14. Our complete haptic augmented reality system is shown in Fig. 5.15.

5.5.1 Friction Actuation Design

One of the advantages of our device over a traditional force-feedback haptic device is its low inherent friction, which was described in Section 5.3. When the tool is dragged across the surface, the lowest friction that can be displayed is the friction in the socket when the ball is allowed to roll freely ($\mu = 0.17$).

To alter the friction between the tool and the surface, we actively change how difficult it is for the ball to roll by applying a force to the top of the rolling ball

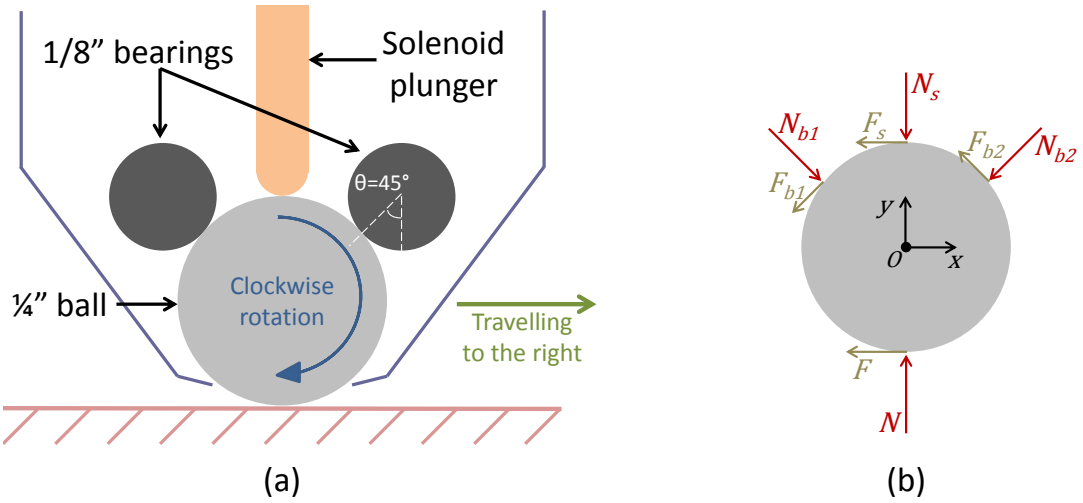

Figure 5.16: (a) Simplified socket design including additional solenoid plunger used in analysis. (b) Free body diagram of forces applied to the rolling ball. The additional friction force applied by the solenoid plunger alters how difficult it is for the ball to roll.

through a rubber-tipped rod, as shown in Fig. 5.16. We embed a linear solenoid in the body of the pen to control the force that the rod applies to the ball, which changes the amount of friction that slows the ball down.

5.5.2 Dynamic Analysis

To analyze the dynamic behavior of this modified socket design, we look at the simplified two-dimensional representation shown in Fig. 5.16(a). The assumptions from Section 5.3.4 are valid in this analysis as well. The free body diagram showing the forces acting on the system is shown in Fig. 5.16(b). We assume the friction between the bearings and the ball follows the linear Coulomb relationships shown in Eq. (5.3.4). Although the center ball is rolling without slipping, we want the user to feel as though the tip of the pen is sliding on the surface with the coefficient of friction

μ. To accomplish this, we enforce the Coulomb relationship shown in Eq. (5.3.6) to create the system's effective surface friction.

Assuming the system is moving at a constant velocity, the force and moment balance equations are:

$$\begin{aligned}\sum F_x = &- \mu N - \mu_b N_{b1}\sin(45^\circ) + N_{b1}\sin(45^\circ) - \mu_s N_s \\ &- \mu_b N_{b2}\sin(45^\circ) - N_{b2}\sin(45^\circ) = 0\end{aligned} \tag{5.5.1}$$

$$\begin{aligned}\sum F_y = &N - \mu_b N_{b1}\cos(45^\circ) - N_{b1}\cos(45^\circ) - N_s \\ &+ \mu_b N_{b2}\cos(45^\circ) - N_{b2}\cos(45^\circ) = 0\end{aligned} \tag{5.5.2}$$

$$\sum M_{z_{@o}} = -\mu N R + \mu_b N_{b1} R + \mu_s N_s R + \mu_b N_{b2} R = 0 \tag{5.5.3}$$

where R is the radius of the ball.

The coefficient of kinetic friction between the solenoid plunger and the ball was experimentally determined to be $\mu_s = 0.6$. Solving Eq. (5.5.1)–(5.5.3) for a desired effective surface friction μ, we find that the force the solenoid plunger must apply to the ball is proportional to the normal force the user applies:

$$N_s = \varphi N \tag{5.5.4}$$

where φ is the scalar parameter:

$$\begin{aligned}\varphi =& \frac{\mu\left(\mu_b^2\left(\sin(45^\circ)+1\right)+\sin(45^\circ)\right)\cos(45^\circ) - \mu_b\sin(45^\circ)}{\mu_b^2\mu_s\left(\sin(45^\circ)-1\right)\cos(45^\circ) - \mu_b\sin(45^\circ) + \mu_s\sin(45^\circ)\cos(45^\circ)} \\ =& \frac{\mu\left(\mu_b^2(\sqrt{2}+1)+1\right)+\sqrt{2}\mu_b}{-\mu_b^2\mu_s\left(\sqrt{2}-1\right)-\sqrt{2}\mu_b+\mu_s}\end{aligned} \tag{5.5.5}$$

The values for μ_b and μ_s are known properties of the system, and μ is the desired

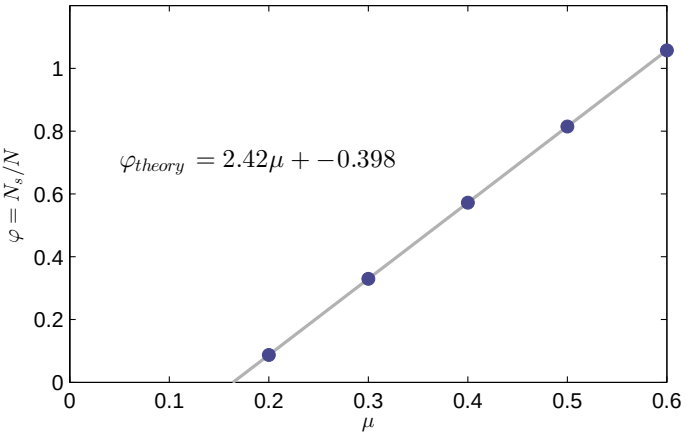

Figure 5.17: Theoretical relationship between the user's normal force and the force required by the solenoid to reach a specified coefficient of friction between the tool and the surface.

effective surface friction.

Fig. 5.17 shows the value of φ as a function of the desired surface friction μ. Although the value of φ is allowed to be greater than one in the theoretical calculations, we do not expect this condition to occur in the physical system. If the force applied by the solenoid N_s is greater than the user's normal force N, the solenoid will push the ball off of the bearings. In this situation, the only forces applied to the ball will be N and N_s, which must be equal in order to satisfy the force balance. Thus, N_s cannot be greater than N in the physical system. Setting the constraint $\varphi \leq 1$, the theoretical maximum surface friction for our prototype is $\mu_{\max} = 0.578$.

This coefficient of friction is higher than the friction of 79 of the 100 modeled textures available in our texture toolkit described in Appendix A. Therefore, the friction range of our device is sufficiently large to match the friction of the majority of the modeled textures.

5.5.3 Characterizing the Solenoid

This Section discusses our analysis of the solenoid and its mechanical and electrical characteristics. In particular, the solenoid's temperature response and force output both affect system performance.

Solenoid Selection

One of the critical characteristics we considered in choosing a solenoid is its size. Since the solenoid must be embedded in the body of the pen, it could not be too large or the tool would be unwieldy. The second critical characteristic we considered is the solenoid's force rating. We needed a solenoid that could apply forces high enough to produce a wide range of braking forces to the ball. If the user applies the maximum normal force we expect in the system ($N = 5.0$ N), the force the solenoid would have to apply to the ball to achieve the theoretical maximum surface friction is 1.73 N. To satisfy these requirements, we chose a 0.5 in. diameter, 1.0 in. long tubular solenoid (Ledex, STA 195203-234).

As shown in Fig. 5.14, the solenoid is placed near the back of the pen. This placement was necessary in order to address concerns about the temperature of the solenoid by including a heatsink. A custom 90 mm long solenoid plunger was machined out of tool steel to accommodate the solenoid's placement and allow it to apply forces to the steel ball at the tip.

Force vs. Current Relationship

The theoretical relationship between current applied to the coil and the force exerted by the solenoid plunger can be determined by first looking at the magnetic field induced in the coil when current runs through the wire. The magnetic field B at the open edge of the solenoid can be calculated as:

$$\begin{aligned} B =& \frac{\mu_0 in}{2}\left(\frac{\ell}{\sqrt{\ell^2 + r^2}}\right) \\ \approx& \frac{\mu_0 in}{2} \quad \text{if } \ell >> r \end{aligned} \tag{5.5.6}$$

where μ_0 is the magnetic permeability of free space, i is the current in the coil, n is the number of coil windings per meter, ℓ is the length of the coil, and r is the radius of the coil. The magnetic permeability is a measurement of the ability of the solenoid plunger to support the formation of a magnetic field within itself.

Through conservation of energy, the force that the coil exerts on the solenoid plunger can be found using the difference in the energy of the magnetic field between when the solenoid is empty and when the plunger is fully inside the solenoid. This change in energy can be calculated as:

$$\begin{aligned} \Delta U =& U_{\text{filled}} - U_{\text{empty}} \\ =& \frac{\ell \pi r^2 B^2 \mu_r}{2\mu_0} - \frac{\ell \pi r^2 B^2}{2\mu_0} \end{aligned} \tag{5.5.7}$$

where $L\pi R^2$ is the volume of the solenoid plunger and μ_r is the relative magnetic permeability of the plunger material. The holding force of the solenoid plunger can

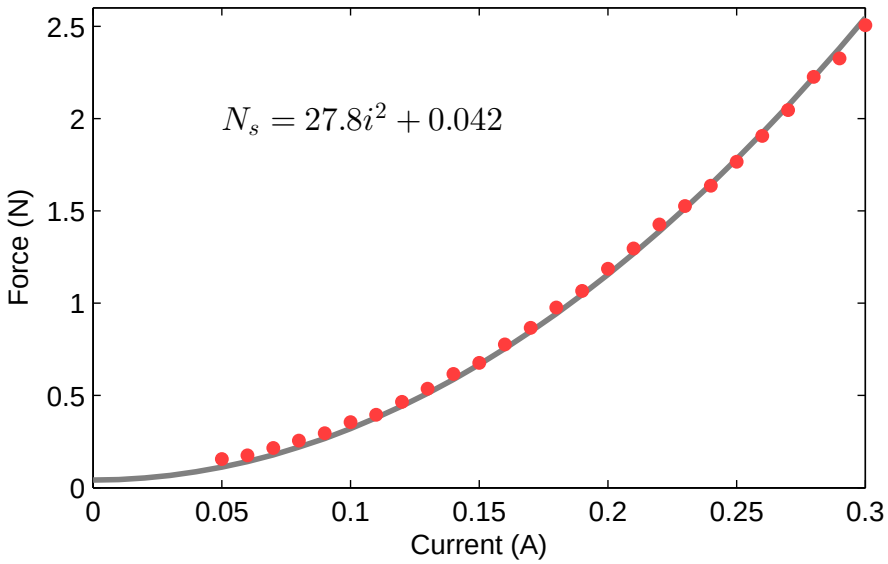

Figure 5.18: Measured and theoretical relationship between force and current in our solenoid. The minimum force that the solenoid can apply is equal to the plunger's weight.

be calculated as:

$$N_s = \frac{\Delta U}{\ell} = \frac{\pi r^2 B^2}{2\mu_0}\,(\mu_r - 1) \tag{5.5.8}$$

The relationship between force and current for our solenoid was experimentally determined by testing a range of current values. For each current value, the body of the solenoid was held while a force was applied to the end of the solenoid plunger. The peak force that was applied before the plunger moved was recorded as the force that the solenoid coil exerted on the plunger. The resulting peak force values are shown in Fig. 5.18. The line shown in Fig. 5.18 is calculated using Eq. (5.5.8). The theory shows a very good fit to the measured force values. The solenoid plunger is made out of tool steel, which has a relative magnetic permeability of $\mu_r = \mu_m/\mu_0 = 2780$. The minimum force that the solenoid can exert on the ball is the weight of the solenoid

plunger, which for our system is 0.042 N. This minimum force could be completely removed by adding a return spring on the solenoid, but this would lower the maximum normal force that could be applied by the solenoid to the ball.

Temperature Analysis

Heat is generated in the solenoid coil when current is applied. To avoid quickly overheating the solenoid, a metal heat sink was added around the solenoid to allow heat to be conducted from the solenoid through the metal tube. Once the heat reaches the outside of the metal tube, it is lost via convection to the ambient air, cooling the device.

To analyze the temperature in the solenoid over time, four levels of constant current were applied for eight minutes each, always starting from room temperature. The level of heat generation is greater than the level of cooling, causing the temperature of the coil to increase over time. This heating results in an increase in the resistance of the coil wire. Although it is difficult to measure the coil temperature directly, we can estimate the temperature by measuring the coil resistance. During each trial the voltage drop across the coil was recorded every 30 seconds. Since we are applying a constant current, the resistance of the coil is linearly related to the measured voltage via Ohm's Law ($R(t) = V(t)/i$). From the measurements of coil resistance, the temperature in the coil can be calculated as:

$$T(t) = T_i + \frac{R(t) - R_i}{\alpha R_i} \tag{5.5.9}$$

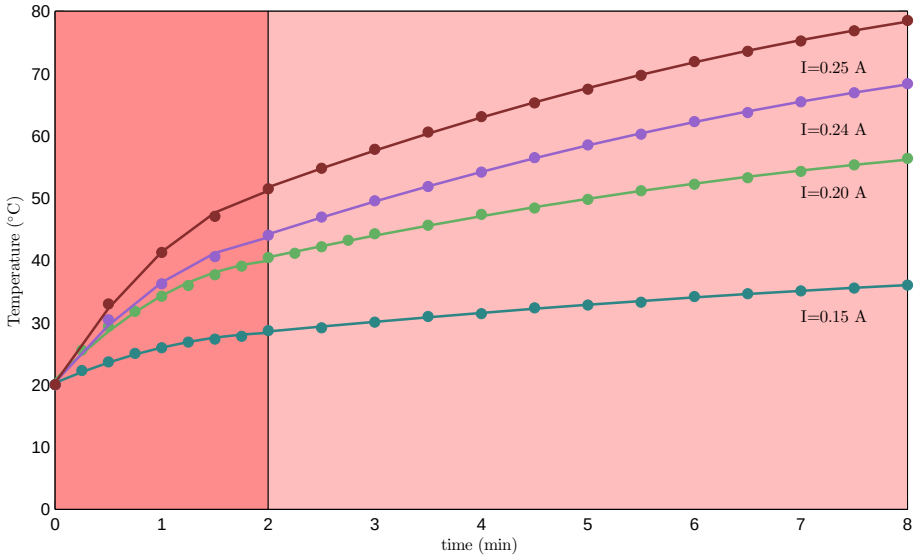

Figure 5.19: Temperature of the solenoid coil over time for different current values. There are two distinct regions of heating in the coil.

where T_i is the ambient temperature, $R(t)$ is the resistance in the coil at time t, R_i is the initial resistance in the coil, and α is the temperature coefficient of resistance for copper ($\alpha = 0.00386\ °C^{-1}$).

Fig. 5.19 shows the estimated temperature of the solenoid coil over time for four levels of constant current. The plot indicates two distinct regions of heating in the solenoid. These different regions, before and after $t = 2$ min, exist because the rate of cooling of the heat sink is greater at higher temperatures. Thus, the rate of temperature increase over time is smaller after the system initially heats up. In both regions the temperature increases approximately quadratically with time.

When driven at 100% duty cycle, the power output by the solenoid should be limited to 4 W to avoid damage due to the temperature of the device. Our chosen solenoid is rated to withstand temperatures up to 120 °C. We did not come close to

reaching this potentially damaging temperature in any of our trials, as shown by the estimated coil temperature in Fig. 5.19. However, it is not enough to consider only the temperature of the coil. We also have to consider the comfort of individuals who will be using our device. The solenoid's heat sink is built as part of the pen. Although it is above the point where the user will grip the pen, it is possible for the heat sink to come into contact with the user's hand. Therefore, we must avoid temperatures that are too high to comfortably touch. This does not necessarily limit the currents that we can use to drive the solenoid, but it may limit the amount of time one can use the system before it becomes uncomfortably hot.

Current Drive vs. Voltage Drive

As shown in Eqs. (5.5.6) and (5.5.8), the force applied by the solenoid is directly related to the current flowing through the coil. Thus, to apply a constant force with the end of the solenoid plunger, the current flowing through the coil must be kept constant. As discussed above, however, as the solenoid heats up over time its resistance increases. Therefore, if a constant voltage was commanded, the current through the coil would decrease over time. To maintain a constant force, we must use current drive instead. By specifying a current through the coil, we can effectively specify the force to be applied by the solenoid plunger to the ball. We use a current amplifier with a gain of 1 A/V, similar to the amplifier used to drive the Haptuator.

5.5.4 Friction Control Scheme

This Section discusses our control scheme for modulating the effective kinetic friction between the tool and the surface.

The analysis in Section 5.5.2 showed that for a desired surface friction μ, the force that must be applied by the solenoid N_s is proportional to the normal force applied by the user N. As shown in Fig. 5.17, this ratio φ is linearly related to the desired surface friction.

Section 5.5.3 determined that the force applied by the solenoid is quadratically related to the commanded current. Thus, if the force that the solenoid should apply to achieve a specified level of surface friction is known, the required current can be calculated using the relationship determined in Fig. 5.18:

$$i = \sqrt{\frac{1}{27.8}\left(N_s - 0.042\right)} \tag{5.5.10}$$

The value of F_s can then be determined using the relationship shown in Eq. (5.5.4). Thus, the current can be calculated as:

$$i = \sqrt{\frac{1}{27.8}\left(\varphi N - 0.042\right)} \tag{5.5.11}$$

The relationship between φ and μ was experimentally determined by recording the tool's normal and friction forces where the current was calculated as in Eq. (5.5.11) for a range of values for φ. A ten-second recording was taken for each value of φ, and a line was fit to a plot of the measured friction force versus normal force. The slope

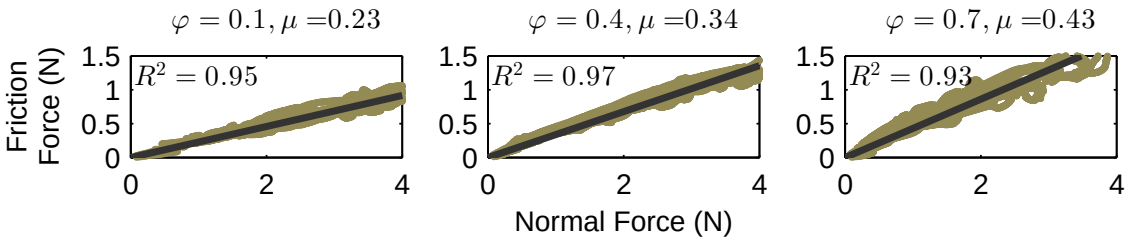

Figure 5.20: Recorded normal and friction force for different values of φ.

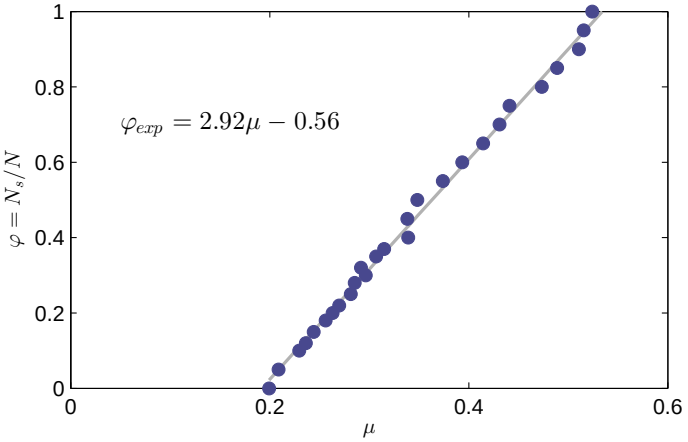

Figure 5.21: Experimental relationship between the user's normal force and the force required by the solenoid to reach a specified coefficient of friction between the tool and the surface.

of this line is the surface's effective coefficient of friction μ. Representative plots of the recorded force values are shown in Fig. 5.20. The plots exhibit a strong linear fit with slight deviation from the average friction value, which illustrates that the surface friction can accurately be controlled by setting φ to be a constant value. If such performance was deemed insufficient, feedback control could be added.

Fig. 5.21 shows the experimental relationship between φ and the resulting surface friction. The correlation is highly linear and closely matches the theoretical

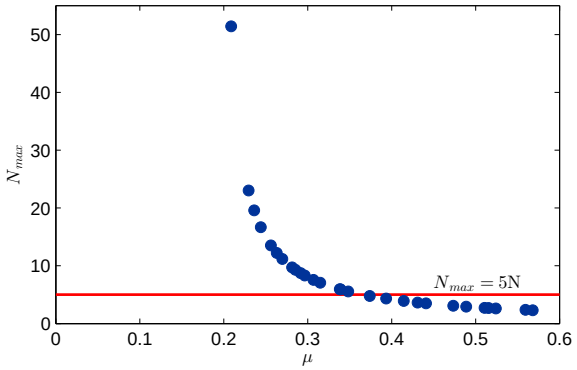

Figure 5.22: Maximum normal force the user can apply as a function of the desired surface friction.

relationship determined in Fig. 5.17. Thus, the solenoid current can be calculated as:

$$i = \sqrt{\frac{1}{27.8}\left((2.92\mu - 0.56)\,N - 0.042\right)} \tag{5.5.12}$$

where N is the tool's measured normal force and μ is the desired coefficient of friction. Therefore, only real-time measurements of the tool's normal force are needed to accurately control the friction between the tool and surface.

The minimum measured coefficient of friction was $\mu_{\min} = 0.20$, which is slightly higher than the friction measured in Section 5.3.5. This discrepancy is caused by the rubber-tipped solenoid plunger, which is always in contact with the ball. The maximum measured coefficient of friction was $\mu_{\max} = 0.52$, which is close to the theoretical maximum determined in Section 5.5.2. However, this maximum friction cannot be achieved for all interaction conditions. To avoid damaging the solenoid by exceeding its maximum rated power, we limit the current to $i_{\max} = 0.3$ A. Fig. 5.22 shows the maximum normal force the user can apply before the current required to drive the solenoid exceeds $i_{\max}$. For coefficients of friction higher than $\mu = 0.35$ N,

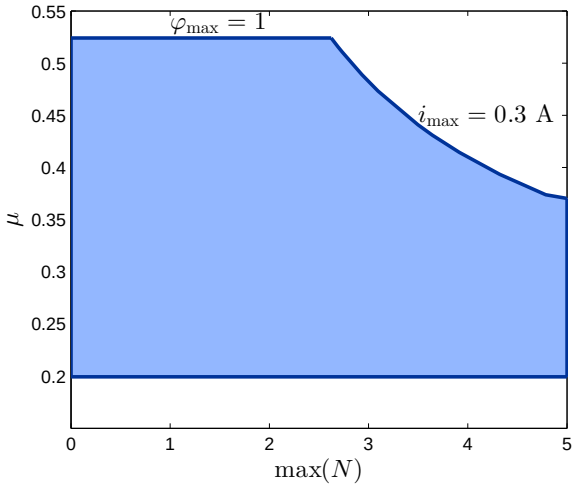

Figure 5.23: Operating range of friction modulation system.

the maximum normal force the user can apply is less than our assumed limit of $N_{\max} = 5$ N. Thus, for higher friction values, the user's normal force should be restricted so the solenoid's current limit is not exceeded. Fig. 5.23 shows the operating range for our friction modulation system.

5.5.5 System Analysis

After determining an appropriate friction control scheme, we wanted to test how well the system performed in outputting the commanded texture vibrations and friction. Fig. 5.24 shows the commanded and measured accelerations and effective friction coefficients for ABS Plastic and Tarp. The commanded and measured vibrations closely match the commanded vibrations in both overall power and variability over time.

A constant friction coefficient was commanded using Eq. (5.5.12) where μ is the

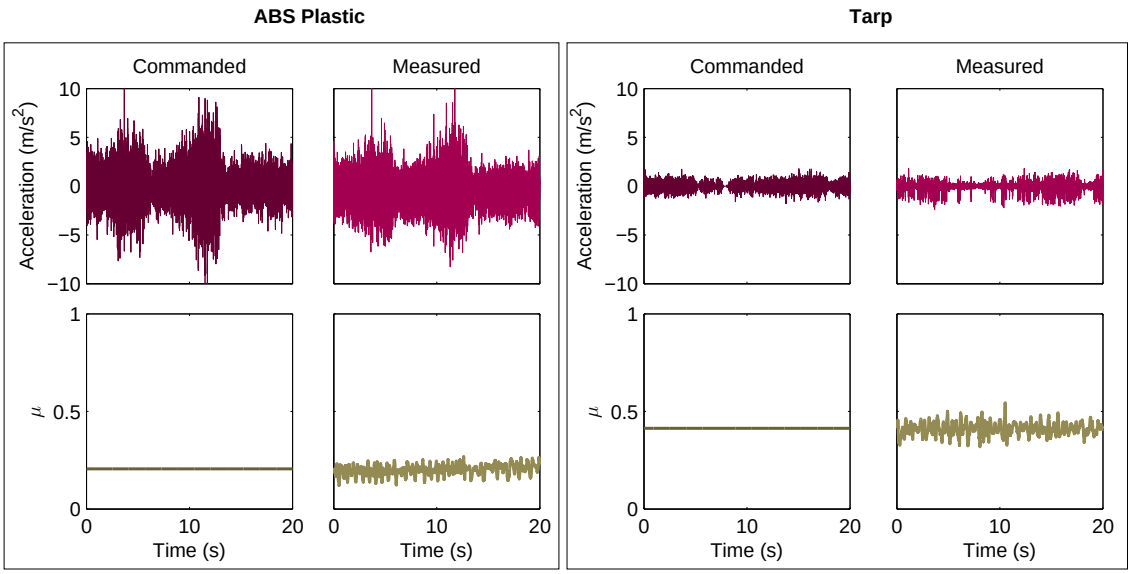

Figure 5.24: Comparison of commanded and measured accelerations and effective friction coefficients for two materials.

real surface's coefficient of friction. The measured friction coefficient was calculated at each time step using the present measurements of normal force and friction force. Both ABS Plastic and Tarp show a measured effective friction coefficient that oscillates around the commanded friction coefficient. Although the measured friction coefficients are not constant, the deviations we see are on the same order of magnitude as variations in measured friction for real surfaces. Therefore, the results in Fig. 5.24 show that the system is able to accurately modulate the effective friction to match the friction of the real surface, while simultaneously displaying realistic friction vibrations.

5.6 Summary

This Chapter presented a novel haptic augmented reality system for altering the perceived texture and friction of a physical three-dimensional object. First we described a method for determining the contact position on a three-dimensional object using a geometric model of the object and measurements of the applied force and torque. Analysis of this method indicated that the position resolution is approximately 0.5 mm for our system, which is comparable to other commercially available position measurement systems. Next, we described the design of a low friction, low vibration pen created to ensure smooth interactions with the physical object. This haptically clean experience is necessary to increase the realism of the additional haptic texture and friction feedback. The design centers on a steel ball supported by four small steel bearings; the ball rolls on the surface without slip. We then rigidly attached a voicecoil actuator to the pen to display previously modeled texture vibrations, and we designed a novel mechanism for modulating the effective friction between the pen and the physical surface. This friction control includes a solenoid that applies a braking force to the ball, increasing the friction in the socket. The braking force translates to an increase in friction between the ball and the surface. The experimental range of friction between the tool and surface was from μ =0.20 to μ =0.52. This range of friction coefficients is sufficient to render a large range of physical surfaces, including 63 of the 100 textures included in our Penn Haptic Texture Toolkit. Our analysis of this system showed that it was able to accurately output

the commanded texture vibrations and modulate the surface friction to a commanded value. This haptic augmented reality system can be used to create a wide array of haptic surfaces, while alleviating the problems of low stiffness and undesired internal friction and vibrations that are common to many force-feedback devices.

Chapter 6

Conclusion

The sensations you feel when interacting with the physical world through a tool are rich and varied. These haptic signals provide you with important information about the texture, friction, and hardness of objects in your surroundings. Fully capturing the complexity of these interactions is essential for creating realistic haptic virtual objects and surfaces. Unfortunately, most haptic rendering schemes fall short of this mark.

This thesis presents methods for modeling haptic virtual surfaces using acceleration, force, and speed data recorded while dragging or tapping on a physical surface. These models are used to create virtual surfaces using three separate rendering systems, which are evaluated for realism.

6.1 Contributions

Recording and modeling of haptic textures

Chapter 2 of this thesis describes new methods for creating haptic texture vibration models from data that was recorded during interactions with a textured surface. One benefit of this new method is that it allows texture data to be recorded without constraints on the normal force applied or the scanning speed used during recording. This unconstrained recording is an advantage over the previous methods described in [95], which required that data be recorded at constant force and speed. However, humans are not robots and cannot maintain perfectly constant force and speed, which resulted in undesired behavior in the acceleration signal in the previous modeling approach. Conversely, our new texture modeling method is robust to changes in the spectrum of the acceleration signal. Instead of creating a single texture model for each data recording, the acceleration signal is broken into segments where the power and spectral properties are approximately constant. The signal is then modeled as a piecewise autoregressive process and the individual autoregressive models are stored in a Delaunay triangulation. This Chapter also presents a new method for stably generating synthetic texture signals from user force and speed values. We tested the recording and modeling process with novice data recorders. The results showed that our methods were robust across different users with minor variations due to differences in hand dynamics or the range of forces and speeds used during recording.

This modeling method, however, is valid for only isotropic and homogeneous textures. Although this represents a large portion of natural textures, a different modeling approach would be needed to model a wider range of textures, including those that are anisotropic or periodic.

Rendering and evaluation of haptic textures

Chapter 3 presents methods for displaying haptic textures on a tablet computer using the texture models that were created in Chapter 2. The tablet and stylus provided measurements of the user's force and speed to generate the texture signals in real time. These signals were then displayed to the user by being played through a voice-coil actuator that was rigidly attached to the stylus. A human-subject study showed that although the textures rendered in this manner were very realistic, we succeeded in capturing only the roughness information of the surface with the vibrations. This modeling and rendering approach is not sufficient to fully capture the real surface's hardness and slipperiness information.

Three-dimenensional rendering of haptic surfaces

Chapter 4 describes a more complete method of creating haptic models of textured surfaces. To overcome the limitations of the vibration-only texture models, additional surface friction and tapping transients were added to more fully capture the surfaces' perceptual qualities. These model components were created from data recorded while

dragging or tapping a tool on real surfaces. These additional model components required force feedback in addition to vibrotactile feedback, which necessitated the switch to a Phantom Omni augmented with a voice-coil actuator for rendering. These haptic surface models were evaluated through two human subject studies to test their realism. The first study investigated whether it is necessary for texture vibrations to respond to both user force and speed during rendering. The results of this study indicated that to create realistic virtual surfaces it is necessary for the texture vibrations to respond to user speed, but not necessarily user force. This result is important to haptics researchers because incorporating force responsiveness might not be feasible or practical due to limitations in texture recording, modeling, or rendering; including speed responsiveness is easier in terms of both computational complexity and hardware. The second study investigated whether it is necessary to include all three model components (surface friction, tapping transients, and texture vibrations) to create realistic surfaces. The results of the study showed that displaying all three components fully captured the surface's slipperiness, hardness, and roughness information. However, it may not be necessary to include all three components to create a realistic virtual surface. The relative importance of the three model components depends on the underlying properties of the real surface.

Haptic augmented reality

Chapter 5 describes a novel haptic augmented reality system for altering the perceived texture and friction of a physical three-dimensional object. Due to size limitations of our device, we do not include force and position sensors in the pen itself. Rather, we offload the sensing by placing a force/torque sensor underneath the physical object. Using only the measurements from this sensor and a three-dimensional geometric model of the object, we accurately calculate the tool's contact position. We created a custom pen for interacting with the physical object. The pen was designed to have low inherent friction and vibrations so as to not contaminate the texture and friction signals that are displayed. We rigidly attached a voicecoil actuator to the pen to display texture vibrations. We then created a novel friction modulation actuator by applying a braking force to a rolling ball in the tip of the pen. This braking force translates to an increase in friction between the tooltip and the surface. Analysis of the system showed that it is capable of accurately modulating the friction of the physical object in the range of μ =0.20 to μ =0.52

6.2 Future Directions

This thesis presents a significant contribution to the field of data-driven haptic modeling and rendering. However, there are limitations to our methods resulting from assumptions we have made concerning the characteristics of the haptic surfaces. The

modeling and rendering methods presented here are valid only for surfaces that are isotropic, which means that they must have the same haptic properties in all directions. This assumption excludes an important group of materials including corduroy and wood with a thick grain. Modeling anisotropic surfaces would require that the recorded signals take into account the direction of the tool.

A second avenue for expansion of these haptic surface models was discussed in Section 4.5.4. In that study, the sample of carpet was not captured well by our surface modeling methods because its surface properties allowed the tool to penetrate the surface and get temporarily stuck during dragging. The resulting lateral impulse forces on the tooltip are important to the feel of the surface, but they are not captured through texture vibrations alone. Rather, fully capturing this behavior would require that the penetration of the tool into the surface and the lateral forces be modeled as a separate component. Capturing this behavior would allow us to model a wider variety of surfaces including soft foam and thick woven fabric in addition to carpet.

A further improvement to the modeling and rendering methods would be to include the sounds generated when the tool is dragged across a surface. In the human subject studies presented in this thesis, we blocked auditory cues by having subjects wear headphones while touching real or virtual surfaces. However, in real interactions haptic and auditory cues work together to form your perception of the world. Furthermore, auditory cues can influence your perception of how an object feels [100]. To fully capture the richness of real objects, virtual rendering should include both

haptic and auditory cues. The modeling methods presented in Chapter 2 could not be directly applied to modeling sound because the autoregressive model structure does not preserve the signal's phase. However, a model interpolation scheme similar to the one described in Section 2.4.1 could be implemented if auditory models were available for a range rendering conditions.

The modeling and rendering principles presented in this thesis could be applied to other haptic applications where realism is desired or a necessity. Increasing haptic realism is especially crucial for medical simulators where doctors train to learn a new skill before performing it on a human patient. Unrealistic haptic feedback is considered by many to hinder widespread adoption of this technology into the medical curriculum [39]. Our data-driven modeling methods could be adapted to model the types of signals doctors experience during procedures such as detecting a cavity or giving an epidural injection. These models could then be added to virtual medical simulators to increase fidelity.

Appendix A

The Penn Haptic Texture Toolkit

As the field of robotics has expanded, there has been a push toward the development of open-source data and software, e.g., [87]. This collaborative environment facilitates the timely advancement of technology because it allows researchers to build upon the work and ideas of their colleagues around the world. Open-source contributions have also flourished in the vision community through the use of vision databases, which provide researchers with a standard set of data with which to test their algorithms. These databases save the research community time because collecting data can be time intensive and expensive.

Several texture databases have been developed for use in computer vision research. The Brodatz repository is a set of black and white images that are considered the classic standard in texture image processing research [110]. The VisTex database contains images of real-world textures under everyday conditions [91]. The MeasTex image

database was compiled to supplement the Brodatz images and the VisTex database sets [99]; it contains a quantitative measurement framework for image texture analysis algorithms. The DynTex database contains sequences of dynamic texture videos [88]. These texture databases allow for the standardization of test conditions and acceleration in the research pipeline. However, there have not yet been any similar databases developed for use in haptics research. This lack of freely available data is probably limiting the speed at which haptic technology can be advanced.

Both our group and others have recently been applying data-driven modeling to haptic textures, with a variety of approaches being explored [24–26, 36, 82, 86, 95, 96, 107]. Our texture modeling approach was discussed in Chapters 2-3. Other researchers have begun asking us to share data collected with our recording system discussed in Section 2.2 and shown in Fig. A.2. Furthermore, at present there is no straightforward way for this group of researchers to compare and validate their modeling and rendering methods. To fulfill this need, we have developed the Penn Haptic Texture Toolkit (HaTT), which consists of haptic texture and friction models of 100 materials, the original data used to create the texture and friction models, images of the materials, and the code and methods necessary to render these textures using a SensAble Phantom Omni. Including the rendering code has the additional benefit of allowing others to incorporate our haptic textures into their virtual environments to create richer and more immersive experiences for their users. For example, a designer could make a tactile virtual room with one of our carpet textures on the floor, our

painted brick texture on the walls, and our wood texture on the door.

Enabling others to test and use our virtual textures required us to modify both our modeling and rendering methods. Our past work discussed in Chapters 2-3 and published in [24,26,95,96] rendered virtual textures on a Wacom tablet using a voice-coil actuator attached to the stylus. The first human-subject study on our textures showed they were the most realistic haptic textures to date [95]. The methods were refined to work with data recorded in an unconstrained manner, Chapter 2, and a second human-subject study was run to evaluate the strengths and weaknesses of this modified approach, Chapter 3. This study found that modeling and rendering textures using only vibrations accurately recreates the roughness of the real textures but is not sufficient to match slipperiness and hardness. Adding in these important perceptual qualities requires a device that can control normal and friction forces, which could not be accomplished using the original TexturePad system. Furthermore, we wanted to make our rendering methods openly available to other researchers, so we chose the widely available SensAble Phantom Omni.

The toolkit is available for public download at `http://repository.upenn.edu/meam\_papers/299/` under copyright from the University of Pennsylvania allowing for the use, copy, and modification of the included software for non-profit research purposes. The rendering code is distributed with permission from Geomagic, the maker of the Omni and OpenHaptics. This Appendix provides a detailed description of the toolkit contents, which has been published in [21].

A.1 Data Collection

This Section describes the specific surfaces that are included in the toolkit and the data that was collected and processed to make the haptic texture and friction models in HaTT. This Section concludes with a description of the recorded data files found in the toolkit.

A.1.1 Texture Samples

The toolkit includes the 100 texture samples shown in Fig. A.1. The materials were chosen to represent different combinations of roughness, hardness, and slipperiness and are divided into ten categories: paper, metal, carbon fiber, fabric, plastic, wood, stone, foam, tile, and carpet. All chosen materials are isotropic and homogeneous, as these conditions are necessary for the modeling methods presented in Chapter 2. Except for metal, all materials were cut to 10.16 cm squares and were mounted on a 0.32-cm-thick piece of acrylic using double-sided tape. The tape was placed only at the edges of the material so it would not affect the compliance of the surface. The toolkit includes close-up color images of all textures, which were taken using a Nikon D40 digital camera. The images are stored as square bitmaps with 1024 pixels on each edge, correlating to a physical scale of 15 pixels/mm.

Figure A.1: Images of the 100 textures that were haptically modeled and are included in the toolkit. The materials are divided into ten categories: paper, metal, carbon fiber, fabric, plastic, wood, stone, foam, tile and carpet. Note that the images are zoomed in from the images included in the toolkit to show detail.

A.1.2 Included Data Files

Data was recorded using the same hardware and methods described in Section 2.2. The hardware, which includes sensors to record the tool's position, orientation, force, and high-frequency acceleration as it is dragged across a textured surface is also shown in Fig. A.2. For this research, a 3.175-mm-diameter stainless steel hemispherical tooltip was attached to the end of the tool and used to interact with the surface. Ten seconds of data were recorded for each texture while the experimenter held the haptic recording device and explored the surface using natural and unconstrained motions. After the data was recorded, it was processed as described in Section 2.2.3.

The toolkit includes two data recordings for each material, which are processed as discussed in Section 2.2.3. The signals are presented at 10 kHz, and all axes are with respect to the magnetic field generator's world frame shown in Fig. A.2. Table A.1 summarizes the included data files, which are stored in XML format.

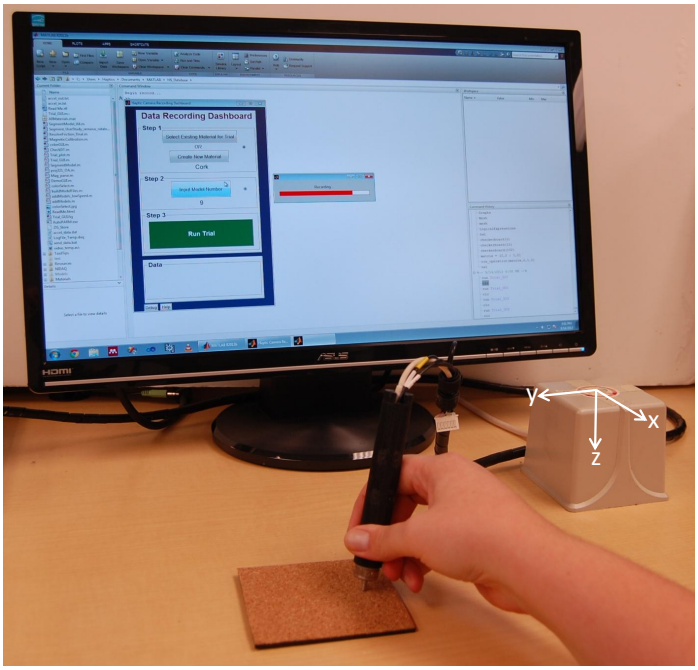

Figure A.2: Haptic recording device. The tool measures position, orientation, force, and acceleration for three axes during interaction with texture samples.

A.2 Haptic Texture and Friction Models

This Section discusses our methods for creating texture and friction models from the acceleration, force, and speed signals recorded in the previous section. We present a new method for resampling the models to render the textures at a sampling rate that differs from the 10 kHz used when recording the data. This Section concludes with a description of the model files that are included in the toolkit.

A.2.1 Texture Models

We model the recorded acceleration signal as a piecewise autoregressive process using the methods presented in Section 2.3. Each AR model is labeled with the median

Table A.1: Recorded Data Files

Field	Description
Accel_*MATERIAL* Files	
<AccelUnits>	Units of acceleration
<SampleRate>	Sample rate in Hz
<Accel>	Combined acceleration signal using DFT321 algorithm
<Accel_x>	Array of acceleration values in x-direction
<Accel_y>	Array of acceleration values in y-direction
<Accel_z>	Array of acceleration values in z-direction
Position_*MATERIAL* Files	
<SpeedUnits>	Units of speed
<PositionUnits>	Units of position
<SampleRate>	Sample rate in Hz
<Speed>	Array of absolute speed values, low-pass filtered at 100 Hz
<Position_x>	Array of position values in x-direction
<Position_y>	Array of position values in y-direction
<Position_z>	Array of position values in z-direction
Force_*MATERIAL* Files	
<ForceUnits>	Units of force
<SampleRate>	Sample rate in Hz
<ForceNormal>	Array of force normal to surface
<ForceTangential>	Array of force values tangential to direction of motion
<Force_x>	Array of force values in x-direction
<Force_y>	Array of force values in y-direction
<Force_z>	Array of force values in z-direction

force and median speed of the segment from which it was made, and these labels are used to store and access the models in a Delaunay triangulation in force-speed space. During rendering of the virtual texture, a texture vibration signal is generated using a measurement of the user's current force and speed, as discussed in Section 2.4.

Model Resampling

The HaTT texture models were created from data recorded at 10 kHz. However, others may want to render the texture vibrations at a sampling rate that is different from that used to record the data. For example, the rendering scheme presented later in this Appendix uses the common haptic sampling rate of 1 kHz. Our desire to share

Table A.2: Model Resampling Files

File	Description
CreateResampledModels.m	Main resampling function. Takes in new sampling rate in Hz as argument.
readValueXML.m	Reads in single value from XML file
readArrayXML.m	Reads in array of values from XML file
resampleModels.m	Takes in the parsed model information, the original sampling rate (10,000 Hz), and the desired sampling rate. Returns the model information of the down-sampled models.
modelXSL_createHTML.xsl	XSL stylesheet to create an HTML file for each new model XML file

these models across platforms and between devices further motivates the need for models at multiple sampling rates. Therefore, we decided to record the data at a much higher sampling rate than needed for our rendering scheme and resample the models to the desired sampling rate. This model resampling creates an equivalent model at a different sampling rate by using a zero-order hold on the inputs. Resampling an AR model adds zeros to the model, which transforms the model to an autoregressive moving-average (ARMA) model, where the output depends on the history of both the past outputs and inputs.

The toolkit includes Matlab code to accomplish this model resampling and write the XML files necessary for the rendering code. Table A.2 provides a summary of the files included. The resampled models must have a sampling rate lower than the original 10 kHz.

A.2.2 Friction Model

A second data recording was made to obtain an estimate of the friction coefficient for the stainless steel hemispherical tooltip dragging across each material. As described in Section 4.2.1, we chose to model this interaction using Coulomb friction, wherein $F_f = \mu_k F_n$. The original recording was found not to be ideal for fitting a simple Coulomb friction model due to the varied input conditions necessary to make a good texture model. Conversely, it was found that a smooth, fluid motion resulted in a more consistent estimate of the friction coefficient because at high speeds the tool glides over the surface [71] so the tangential force is not dominated by impacts with the surface features.

We downsampled the force data to 240 Hz before using the normal and tangential unit vectors to calculate the normal and friction forces. A Coulomb friction model was fit to the data by constraining a line to intersect the origin. The friction coefficients of the surfaces in the toolkit range in value from 0.11 for wax paper to 1.39 for ethylene propylene diene monomer (EPDM) foam.

A.2.3 Included Model Files

The toolkit includes both the original AR models at 10 kHz and the resampled ARMA models at 1 kHz. Table A.3 shows the fields included in the model XML files. The model XML files are used for rendering in the next section. HTML files are also included in the toolkit for visualization of the model sets.

Table A.3: Model Files

File	Description
<Material>	Texture name
<AccelUnits>	Units of acceleration
<SpeedUnits>	Units of speed
<ForceUnits>	Units of force
<SampleRate>	Model sample rate in Hz
<mu>	Kinetic friction coefficient
<htmlPicture>	File path to image of texture for HTML
<renderPicture>	File path to image of texture for rendering
<numMod>	Number of models
<numTri>	Number of triangles in Delaunay triangulation
<maxSpeed>	Maximum modeled speed
<maxForce>	Maximum modeled force
<speedList>	Array of all modeled speeds
<forceList>	Array of all modeled forces
<tri>	Model vertices for one triangle
<DTpicture>	File path to image of Delaunay triangulation
<numARCoeff>	Number of AR coefficients
<numMACoeff>	Number of MA coefficients
<model>	Information for one ARMA model, including:
<modNum>	Model number
<ARcoeff>	Array of AR coefficients
<ARlsf>	Array of AR line spectral frequencies
<MAcoeff>	Array of MA coefficients
<MAlsf>	Array of MA line spectral frequencies
<var>	Model variance
<gain>	Gain of MA coefficients
<speedMod>	Model speed
<forceMod>	Model force

A.3 Texture Rendering

This Section describes the methods used to render the textures and display them to a user. First we describe the hardware, and then we describe the software that handles the rendering calculations and displays the output forces to the user. This Section concludes with a description of the files included in the toolkit necessary for rendering the haptic virtual textures.

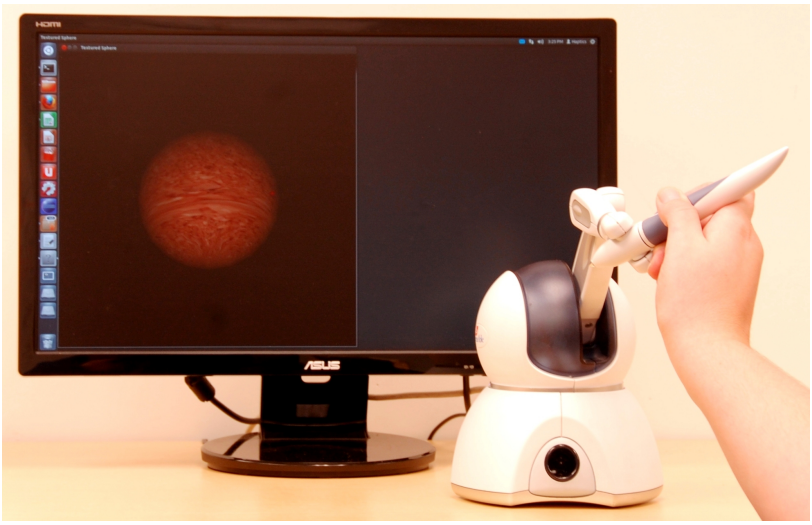

Figure A.3: A person using a Phantom Omni to interact with the cork texture on a virtual sphere.

A.3.1 Hardware

The code presented in the toolkit is for the implementation of our texture rendering methods with a SensAble Phantom Omni. An Omni was chosen because of its low cost and wide availability among haptics researchers. Several researchers have adapted the rendering code to work with other hardware including the Magnetic Levitation Haptic Interface by Butterfly Haptics. The rendering methods presented in the toolkit were originally implemented on a Linux computer running Ubuntu version 12.04 LTS with a GeForce GTX 570/PCIe/SSE2 graphics card. The Windows version of the rendering code has also been implemented in Visual Studio 2008 on a computer running 64-bit Windows 7 with an Intel HD Graphics 2000 graphics card.

A.3.2 Software

The rendering code presented in this toolkit is based on OpenHaptics 3.0 and uses the Haptic Device API (HDAPI), which runs the haptic servo loop thread at 1000 Hz [46]. The included texture rendering program maps the haptic textures on to a 100-mm-diameter sphere, as shown in Fig. A.3. The user interacts with the textured sphere through a 3.2-mm-diameter spherical proxy. Any of the 100 textures can be selected to render from a menu, and the image of the texture is displayed on the sphere.

During rendering, three separate forces are calculated and displayed to the user: the normal force ($\vec{F}_n$), which provides the general shape and hardness of the sphere, the friction force ($\vec{F}_f$), which controls the slipperiness of the surface, and the texture force ($\vec{F}_t$), which adds our modeled textures to the surface.

If a collision between the proxy and the sphere is detected, the software gets the user's current applied normal force and current position. The normal force follows a Hooke's law relationship to the proxy's penetration depth with a gain of 0.5 N/mm. The time history of position is used to estimate the user's current velocity. A second-order low-pass filter at 20 Hz is applied to the velocity estimate to reduce noise and remove movement caused by displaying the texture. We then subtract the component of the velocity vector in the normal direction to obtain the user's velocity tangent to the surface.

The user's tangential velocity vector is used to calculate the unit vector tangent to the direction of motion for the display of friction. Our friction rendering approach

was set by the modeling approach made in Section A.2.2. However, by itself Coulomb friction cannot be used in rendering because there is a discontinuity in the magnitude and direction of the friction force at zero velocity. This discontinuity creates an unrealistic and jarring feeling and can cause the system to become unstable for large friction coefficients. Therefore, it was necessary to add a region of viscous damping around zero velocity to remove discontinuities in the friction force, as shown in Fig. A.4. We aimed to keep this viscous region as small as possible, so the threshold velocity for the transition between viscous damping and Coulomb friction was not kept constant for all materials. Instead we made this threshold velocity a function of the material's friction coefficient:

$$v_{th} = \frac{\mu_k}{b} \tag{A.3.1}$$

where v_{th} is the velocity threshold, and b is the slope of the viscous region divided by the normal force, which was empirically chosen as 0.004 s/mm. Although some of the instability seen in friction rendering could be fixed by switching to a device with a higher position resolution such as the Phantom Premium, we chose to test the rendering code on a Phantom Omni to make the toolkit available to a wider audience. Researchers using the toolkit may wish to modify the friction rendering approach for their own system and could also fit more sophisticated friction models from the provided data.

Using our estimates of the user's current normal force and tangential speed, we synthesize vibrations at 1000 Hz as described in Section 2.4. This signal generation

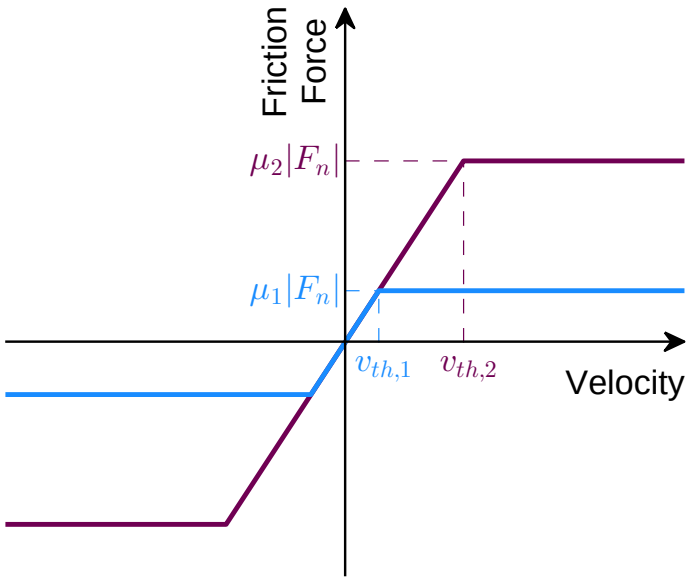

Figure A.4: Friction model used for rendering. A region of viscous damping was added around zero velocity for stability. The slope of the viscous region was kept constant for all materials.

returns accelerations in the units of m/s^2. However, the Omni needs to display forces, so we scale this acceleration by the effective mass of the handle and the user's hand, which was empirically chosen to be 0.05 kg; this value would need to be adjusted for realistic rendering with other haptic interfaces. Since humans cannot discriminate the direction of high-frequency vibration [9], it does not matter in which direction we display the texture forces. Therefore, to minimize the chances of exceeding the output capabilities of the Omni, we create a texture unit vector ($\hat{e}_t$) that is perpendicular to both the normal vector ($\hat{e}_n$) and the friction vector ($\hat{e}_f$), as shown in Fig. A.5.

Each time through the servo loop, if collision with the sphere is detected, the normal, friction, and texture force vectors are calculated, and their resultant sum is displayed to the user. A sample of the three output forces and the user's speed is

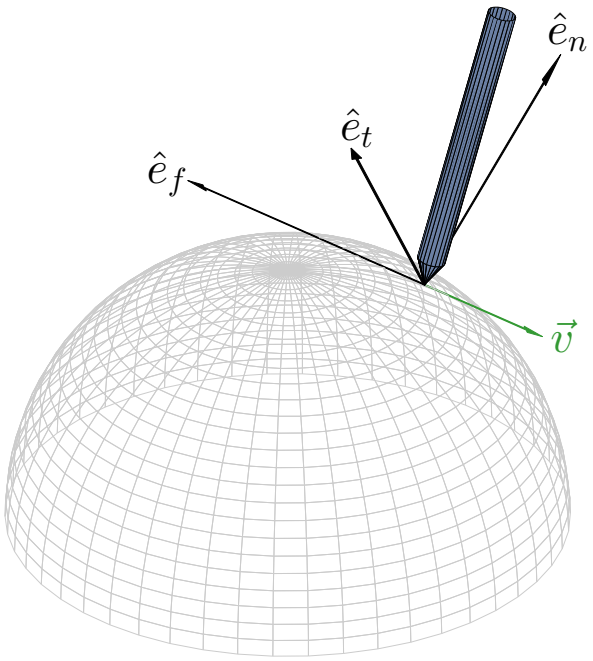

Figure A.5: Direction of forces displayed to the user. $\vec{v}$ indicates the user's direction of travel.

shown in Fig. A.6.

A.3.3 Rendering Code

The files necessary to run the described rendering code are shown in Table A.4. Third party code is explicitly labeled. In addition, the Boost Random Number Library is needed to generate the excitation signals [29].

A.4 Summary

Motivated by the current lack of a freely available database of haptic textures, we have created the Penn Haptic Texture Toolkit of 100 textures. For each texture,

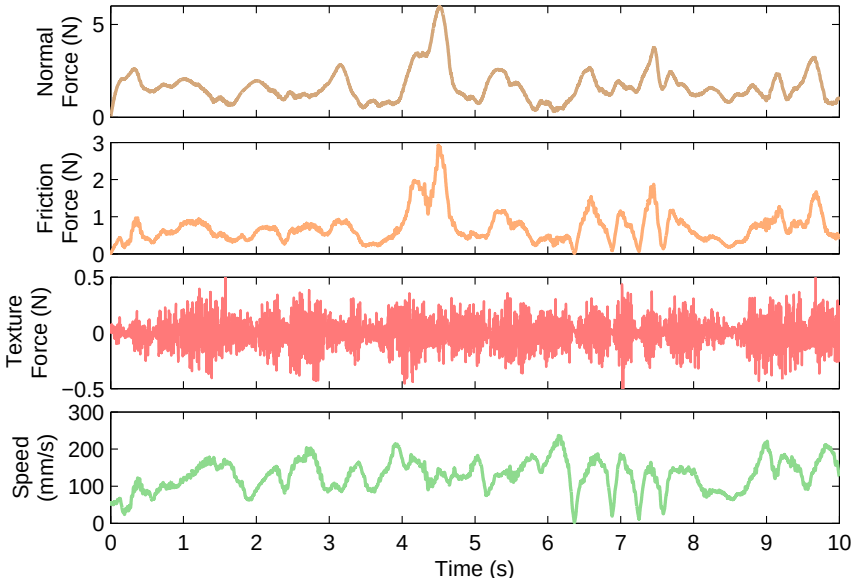

Figure A.6: The three force signals output to the user by the Omni and the speed of the user while interacting with virtual cork.

the toolkit contains two recorded datasets, texture and friction models, an image, and the code necessary to render the textures using a SensAble Phantom Omni. We hope that this toolkit will become a tool to help researchers enhance their haptic virtual environments and also compare and validate texture modeling approaches. This Appendix presents our methods for modeling and rendering haptic virtual textures, as well as the information included in the toolkit. Each texture's model set is created from a ten-second high-frequency acceleration recording, which is segmented and modeled as a piecewise autoregressive process. The individual AR models are labeled with the force and speed used during data recording and are stored in a Delaunay triangulation. A measurement of the user's current normal force and tangential speed can be used to synthesize textures in real time. A method for resampling the models was presented to facilitate the sharing of the models amongst researchers and across

Table A.4: Rendering Files

Field	Description
***src* folder**	
main.cpp	Runs graphics loop and haptics loop. Calculates texture vibrations
AccSynthHashMatrix.cpp	Constructs hash entries to store models. Interpolates between models
ContactModel.cpp*	Defines equation of the force fields
helper.cpp*	Sets the graphics state
pugixml.cpp§	XML parser
***include* folder**	
AccSynthHashMatrix.h	Hash table class for texture generation
autoGenHashMatrix.h	Parses model parameters from xml file and stores models in memory
shared.h	Shared memory file for texture generation
sharedInit.h	Initializes variables in shared memory for texture generation
ContactModel.h*	Defines constants for contact in simulation
Constants.h*	Defines physical constants of simulation
pugixml.hpp§	XML parser
pugiconfig.hpp§	Configuration file for XML parser
foreach.hpp§	Boost.Foreach support for pugixml classes
***build* folder**	
TexturedSphere	Executable for rendering
Models_*MATERIAL*.xml	Model XML files
***images* folder**	
*MATERIAL*_square.bmp	Texture image for display in rendering

*Adapted from OpenHaptics sample code, §Copyright 2006-2012 by Arseny Kapoulkine

devices.

Additionally, the use of the Phantom Omni allowed for the direct rendering of friction, which was shown in Chapter 3 to be a limiting factor in the realism of our textures displayed on a tablet. Chapter 4 presents an updated version of this rendering system, which includes augmenting the Omni's handle with a voice-coil actuator to directly output the texture vibrations instead of using the Omni's motors. This modification allows us to better control the output, which results in higher-fidelity recreations of the synthesized texture vibrations.

Bibliography

[1] *PortAudio - Cross-Platform Audio API*, `http://www.portaudio.com/`, April 2010.

[2] *The Linux Wacom Project*, `http://linuxwacom.sourceforge.net/`, April 2010.

[3] 3D Systems, 430 Davis Drive, Suite 300, Morrisville, NC 27560, *Phantom omni device guide*, April 2013.

[4] Jacopo Aleotti, Francesco Denaro, and Stefano Caselli, *Object manipulation in visuo-haptic augmented reality with physics-based animation*, Proc. IEEE International Symposium on Robot and Human Interactive Communication, 2010, pp. 38–43.

[5] Shuhei Asano, Shogo Okamoto, and Yoji Yamada, *Vibrotactile stimulation to increase and decrease texture roughness*, IEEE Transactions on Human-Machine Systems **45** (2015), no. 3, 393–398.

[6] Cagatay Basdogan, Chih-Hao Ho, and Mandayam A. Srinivasan, *A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments*, Proc. ASME Dynamics Systems and Control Division, 1997, pp. 77–84.

[7] Olivier Bau and Ivan Poupyrev, *REVEL: Tactile feedback technology for augmented reality*, ACM Transactions on Graphics **31** (2012), no. 4, 89.

[8] Benjamin Bayart, Jean-Yves Didier, and Abderrahmane Kheddar, *Force feedback virtual painting on real objects: A paradigm of augmented reality haptics*, Proc. EuroHaptics Conference, Springer, 2008, pp. 776–785.

[9] Jonathan Bell, Stanley Bolanowski, and Mark H. Holmes, *The structure and function of Pacinian corpuscles: A review*, Progress in Neurobiology **42** (1994), no. 1, 79–128.

[10] Sliman J. Bensmaïa and Mark Hollins, *The vibrations of texture*, Somatosensory & Motor Research **20** (2003), no. 1, 33–43.

[11] Wouter M. Bergmann Tiest, *Tactual perception of material properties*, Vision Research **50** (2010), no. 24, 2775–2782.

[12] Wouter M. Bergmann Tiest and Astrid M. L. Kappers, *Analysis of haptic perception of materials by multidimensional scaling and physical measurements of roughness and compressibility*, Acta Psychologica **121** (2006), no. 1, 1–20.

[13] Wouter M. Bergmann Tiest and Astrid M. L. Kappers, *Haptic and visual perception of roughness*, Acta Psychologica **124** (2007), no. 2, 177–189.

[14] Gerald Bianchi, Benjamin Knoerlein, Gabor Szekely, and Matthias Harders, *High precision augmented reality haptics*, Proc. EuroHaptics Conference, vol. 6, 2006, pp. 169–178.

[15] Antonio Bicchi, J Kenneth Salisbury, and David L Brock, *Contact sensing from force measurements*, The International Journal of Robotics Research **12** (1993), no. 3, 249–262.

[16] George E. P. Box and Gwilym M. Jenkins, *Time series analysis: Forecasting and control*, Holden-Day, San Francisco, 1970.

[17] Gianni Campion and Vincent Hayward, *Fundamental limits in the rendering of virtual haptic textures*, Proc. IEEE World Haptics Conference, 2005, pp. 263–270.

[18] Steven A. Cholewiak, Kwangtaek Kim, Hong Z. Tan, and Bernard D. Adelstein, *A frequency-domain analysis of haptic gratings*, IEEE Transactions on Haptics **3** (2010), no. 1, 3–14.

[19] Heather Culbertson and Katherine J. Kuchenbecker, *Data-driven haptic virtual surfaces with friction, hardness, and texture*, Submitted to *IEEE Transactions on Haptics*.

[20] Heather Culbertson and Katherine J. Kuchenbecker, *Should haptic texture vibrations respond to user force and speed?*, Proc. IEEE World Haptics Conference, June 2015.

[21] Heather Culbertson, Juan José López Delgado, and Katherine J. Kuchenbecker, *One hundred data-driven haptic texture models and open-source methods for rendering on 3D objects*, Proc. IEEE Haptics Symposium, IEEE, 2014, pp. 319–325.

[22] ———, *The Penn Haptic Texture Toolkit for modeling, rendering, and evaluating haptic virtual textures*, `http://repository.upenn.edu/meam_papers/299/`, February 2014.

[23] Heather Culbertson, Craig G. McDonald, Benjamin E. Goodman, and Katherine J. Kuchenbecker, *Data-driven modeling and rendering of isotropic textures*, Hands-on Demonstration presented at IEEE World Haptics Conference, Seoul, South Korea, April 2013.

[24] Heather Culbertson, Joseph M. Romano, Pablo Castillo, Max Mintz, and Katherine J. Kuchenbecker, *Refined methods for creating realistic haptic virtual textures from tool-mediated contact acceleration data*, Proc. IEEE Haptics Symposium, March 2012, pp. 385–391.

[25] Heather Culbertson, Juliette Unwin, Benjamin E. Goodman, and Katherine J. Kuchenbecker, *Generating haptic texture models from unconstrained*

tool-surface interactions, Proc. IEEE World Haptics Conference, April 2013, pp. 295–300.

[26] Heather Culbertson, Juliette Unwin, and Katherine J. Kuchenbecker, *Modeling and rendering realistic textures from unconstrained tool-surface interactions*, IEEE Transactions on Haptics **7** (2014), no. 3, 381–393.

[27] Phil R. Dahl, *A solid friction model*, Tech. report, DTIC Document, 1968.

[28] Richard A. Davis, Thomas C. M. Lee, and Gabriel A. Rodriguez-Yam, *Structural break estimation for nonstationary time series models*, Journal of the American Statistical Association **101** (2006), no. 473, 223–239.

[29] Beman Dawes, David Abrahams, and Rene Rivera, *Boost C++ libraries*, URL http://www.boost.org (2009).

[30] Olivier Devillers, Sylvain Pion, and Monique Teillaud, *Walking in a triangulation*, Proc. ACM Symposium on Computational Geometry, 2001, pp. 106–114.

[31] Nicola Diolaiti, Günter Niemeyer, Federico Barbagli, and J. Kenneth Salisbury, *Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects*, IEEE Transactions on Robotics **22** (2006), no. 2, 256–268.

[32] Jonathan P. Fiene and Katherine J. Kuchenbecker, *Shaping event-based haptic transients via an improved understanding of real contact dynamics*, Proc. IEEE World Haptics Conference, IEEE, 2007, pp. 170–175.

[33] Jeremy A. Fishel and Gerald E. Loeb, *Bayesian exploration for intelligent identification of textures*, Frontiers in Neurorobotics **6** (2012).

[34] Jason P. Fritz and Kenneth E. Barner, *Stochastic models for haptic texture*, Proc. SPIE International Symposium on Intelligent Systems and Advanced Manufacturing, 1996, pp. 34–44.

[35] Elamir S. Gadelmawla, *Estimation of surface roughness for turning operations using image texture features*, Proc. of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **225** (2011), no. 8, 1281–1292.

[36] Vijaya L. Guruswamy, Jochen Lang, and Won-Sook Lee, *Modeling of haptic vibration textures with infinite-impulse-response filters*, Proc. IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE), November 2009, pp. 105–110.

[37] Taku Hachisu, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto, *Augmentation of material property by modulating vibration resulting from tapping*, Haptics: Perception, Devices, Mobility, and Communication, Springer, 2012, pp. 173–180.

[38] Blake Hannaford and Allison M. Okamura, *Haptics*, Springer handbook of robotics (B. Siciliano and O. Lhatib, eds.), Springer, 2008, pp. 719–739.

[39] Matthias Harders, *Haptics in medical applications*, Haptic Rendering: Foundations, Algorithms, and Applications (M. Lin and M. Otaduy, eds.), AK Peters, 2008, pp. 501–515.

[40] Matthias Harders, Gérald Bianchi, Benjamin Knoerlein, and Gábor Székely, *Calibration, registration, and synchronization for high precision augmented reality haptics*, IEEE Transactions on Visualization and Computer Graphics **15** (2009), no. 1, 138–149.

[41] Paul A. Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, and Jose G. Conde, *Research electronic data capture (REDCap)-A metadata-driven methodology and workflow process for providing translational research informatics support*, Journal of Biomedical Informatics **42** (2009), no. 2, 377.

[42] Vincent Hayward and Brian Armstrong, *A new computational model of friction applied to haptic rendering*, Experimental Robotics VI, Springer, 2000, pp. 403–412.

[43] Hsin-Ni Ho and Lynette A. Jones, *Contribution of thermal cues to material discrimination and localization*, Perception & Psychophysics **68** (2006), no. 1, 118–128.

[44] ———, *Development and evaluation of a thermal display for material identification and discrimination*, ACM Transactions on Applied Perception **4** (2007),

no. 2, 13.

[45] Mark Hollins, Richard Faldowski, Suman Rao, and Forrest Young, *Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis*, Attention, Perception, & Psychophysics **54** (1993), no. 6, 697–705.

[46] Brandon Itkowitz, Josh Handley, and Weihang Zhu, *The OpenHaptics toolkit: a library for adding 3D touch navigation and haptics to graphics applications*, Proc. IEEE World Haptics Conference, 2005, pp. 590–591.

[47] Seokhee Jeon and Seungmoon Choi, *Haptic augmented reality: Taxonomy and an example of stiffness modulation*, Presence: Teleoperators and Virtual Environments **18** (2009), no. 5, 387–408.

[48] ———, *Stiffness modulation for haptic augmented reality: Extension to 3D interaction*, Proc. IEEE Haptics Symposium, IEEE, 2010, pp. 273–280.

[49] ———, *Real stiffness augmentation for haptic augmented reality*, Presence: Teleoperators and Virtual Environments **20** (2011), no. 4, 337–370.

[50] Seokhee Jeon, Seungmoon Choi, and Matthias Harders, *Rendering virtual tumors in real tissue mock-ups using haptic augmented reality*, IEEE Transactions on Haptics **5** (2012), no. 1, 77–84.

[51] Seokhee Jeon, Jean-Claude Metzger, Seungmoon Choi, and Matthias Harders, *Extensions to haptic augmented reality: Modulating friction and weight*, Proc. IEEE World Haptics Conference, IEEE, 2011, pp. 227–232.

[52] Roland S. Johansson and J. Randall Flanagan, *Coding and use of tactile signals from the fingertips in object manipulation tasks*, Nature Reviews Neuroscience **10** (2009), no. 5, 345–359.

[53] Kenneth O. Johnson, *The roles and functions of cutaneous mechanoreceptors*, Current opinion in neurobiology **11** (2001), no. 4, 455–461.

[54] Kenneth O. Johnson, Takashi Yoshioka, and Francisco Vega-Bermudez, *Tactile functions of mechanoreceptive afferents innervating the hand*, Journal of Clinical Neurophysiology **17** (2000), no. 6, 539–558.

[55] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, et al., *Principles of neural science*, vol. 4, McGraw-Hill New York, 2000.

[56] Justin H. Killebrew, Sliman J. Bensmaia, John F. Dammann, Peter Denchev, Steven S. Hsiao, James C. Craig, and Kenneth O. Johnson, *A dense array stimulator to generate arbitrary spatio-temporal tactile stimuli*, Journal of neuroscience methods **161** (2007), no. 1, 62–74.

[57] Chih-Hung King, Martin O. Culjat, Miguel L. Franco, James W. Bisley, Gregory P. Carman, Erik P. Dutson, and Warren S. Grundfest, *A multielement tac-*

tile feedback system for robot-assisted minimally invasive surgery, IEEE Transactions on Haptics **2** (2009), no. 1, 0052–56.

[58] Roberta L. Klatzky and Susan J. Lederman, *Tactile roughness perception with a rigid link interposed between skin and surface*, Perception & psychophysics **61** (1999), no. 4, 591–607.

[59] Roberta L. Klatzky, Susan J. Lederman, Cheryl Hamilton, Molly Grindley, and Robert H. Swendsen, *Feeling textures through a probe: Effects of probe and surface geometry and exploratory factors*, Attention, Perception and Psychophysics **65** (2003), no. 4, 613–631.

[60] Ahmet M. Kondoz, *Digital speech: Coding for low bit rate communication systems*, John Wiley and Sons, Ltd., 2004.

[61] Dimitrios A. Kontarinis and Robert D. Howe, *Tactile display of vibratory information in teleoperation and virtual environments*, Presence: Teleoperators and Virtual Environments **4** (1995), no. 4, 387–402.

[62] Katherine J. Kuchenbecker, Jonathan P. Fiene, and Günter Niemeyer, *Improving contact realism through event-based haptic feedback*, IEEE Transactions on Visualization and Computer Graphics **12** (2006), no. 2, 219–230.

[63] Katherine J. Kuchenbecker, Joseph M. Romano, and William McMahan, *Haptography: Capturing and recreating the rich feel of real surfaces*, Proc. International Symposium of Robotics Research, 2009, pp. 245–260.

[64] Ki-Uk Kyung, Jun-Young Lee, Junseok Park, and Mandayam A. Srinivasan, *Wubi-pen: Sensory feedback stylus interacting with graphical user interface*, Presence: Teleoperators and Virtual Environments **21** (2012), no. 2, 142–155.

[65] Robert H. LaMotte, *Softness discrimination with a tool*, Journal of Neurophysiology **83** (2000), no. 4, 1777–1786.

[66] Nils Landin, Joseph M. Romano, William McMahan, and Katherine J. Kuchenbecker, *Dimensional reduction of high-frequency accelerations for haptic rendering*, Proc. EuroHaptics Conference, July 2010, pp. 79–86.

[67] Susan J. Lederman, *Tactile roughness of grooved surfaces: The touching process and effects of macro- and microsurface structure*, Perception and Psychophysics **16** (1974), no. 2, 385–395.

[68] Susan J. Lederman and Susan G. Abbott, *Texture perception: studies of intersensory organization using a discrepancy paradigm, and visual versus tactual psychophysics.*, Journal of Experimental Psychology: Human perception and performance **7** (1981), no. 4, 902.

[69] Susan J. Lederman and Roberta L. Klatzky, *Action for perception: Manual exploratory movements for haptically processing objects and their features.*, Hand and brain: Neurophysiology and psychology of hand (P. Haggard A. Wing and R. Flanagan, eds.), Academic Press, 1996, pp. 431–446.

[70] Susan J. Lederman, Roberta L. Klatzky, Cheryl Hamilton, and Molly Grindley, *Perceiving surface roughness through a probe: Effects of applied force and probe diameter*, Proc. ASME Dynamics Systems and Control Division, 2000, pp. 1065–1071.

[71] Susan J. Lederman, Roberta L. Klatzky, Cheryl L. Hamilton, and Gillian I. Ramsay, *Perceiving roughness via a rigid probe: Psychophysical effects of exploration speed and mode of touch*, Haptics-e **1** (1999), no. 1.

[72] Susan J. Lederman, Roberta L. Klatzky, Timothy Morgan, and Cheryl Hamilton, *Integrating multimodal information about surface texture via a probe: relative contributions of haptic and touch-produced sound sources*, Proc. IEEE Haptics Symposium, IEEE, 2002, pp. 97–104.

[73] Daniel Liberzon, *Switching in systems and control*, Springer, 2003.

[74] Jack M. Loomis, *Distal attribution and presence*, Presence: Teleoperators and Virtual Environments **1** (1992), no. 1, 113–119.

[75] Karon E. MacLean, *Haptic interaction design for everyday interfaces*, Reviews of Human Factors and Ergonomics **4** (2008), no. 1, 149–194.

[76] Thomas H. Massie and J. Kenneth Salisbury, *The PHANToM haptic interface: A device for probing virtual objects*, Proc. ASME winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, vol. 55, 1994, pp. 295–300.

[77] Craig McDonald and Katherine J. Kuchenbecker, *Dynamic simulation of tool-mediated texture interaction*, Proc. IEEE World Haptics Conference, April 2013, pp. 307–312.

[78] William McMahan and Katherine J. Kuchenbecker, *Dynamic modeling and control of voice-coil actuators for high-fidelity display of haptic vibrations*, Proc. IEEE Haptics Symposium, February 2014.

[79] William McMahan, Joseph M. Romano, Amal M. Abdul Rahuman, and Katherine J. Kuchenbecker, *High frequency acceleration feedback significantly increases the realism of haptically rendered textured surfaces*, Proc. IEEE Haptics Symposium, IEEE, 2010, pp. 141–148.

[80] Margaret Minsky, Ming Ouh-Young, Oliver Steele, Frederick P. Brooks Jr., and Max Behensky, *Feeling and seeing: issues in force display*, Proc. Symposium on Interactive 3D Graphics, vol. 24, 1990, pp. 235–241.

[81] Shogo Okamoto, Hikaru Nagano, and Yoji Yamada, *Psychophysical dimensions of tactile perception of textures*, IEEE Transactions on Haptics **6** (2012), no. 1, 81–93.

[82] Allison M. Okamura, Jack T. Dennerlein, and Robert D. Howe, *Vibration feedback models for virtual environments*, Proc. IEEE International Conference on Robotics and Automation, May 1998, pp. 674–679.

[83] Allison M. Okamura, Katherine J. Kuchenbecker, and Mohsen Mahvash, *Measurement-based modeling for haptic display*, Haptic Rendering: Foundations, Algorithms, and Applications (M. Lin and M. Otaduy, eds.), AK Peters, 2008, pp. 443–467.

[84] Allison M. Okamura, Robert J. Webster III, Jason T. Nolin, K. W. Johnson, and H. Jafry, *The haptic scissors: cutting in virtual environments*, Proc. IEEE International Conference on Robotics and Automation, vol. 1, 2003, pp. 828–833.

[85] Miguel A. Otaduy and Ming C. Lin, *Rendering of textured objects*, Haptic Rendering: Foundations, Algorithms, and Applications (M. Lin and M. Otaduy, eds.), AK Peters, 2008, pp. 371–393.

[86] Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau, *Scanning physical interaction behavior of 3D objects*, Proc. Conference on Computer Graphics and Interactive Techniques, 2001, pp. 87–96.

[87] Caroline Pantofaru, Sachin Chitta, Brian Gerkey, Radu Rusu, William D. Smart, and Richard Vaughan, *Special issue on open source software-supported robotics research*, Autonomous Robots **34** (2013), no. 3.

[88] Renaud Péteri, Sándor Fazekas, and Mark J. Huiskes, *DynTex: A comprehensive database of dynamic textures*, Pattern Recognition Letters **31** (2010),

no. 12, 1627–1632.

[89] Delphine Picard, Catherine Dacremont, Dominique Valentin, and Agnes Giboreau, *Perceptual dimensions of tactile textures*, Acta Psychologica **114** (2003), no. 2, 165–184.

[90] Matthew Piccoli and Mark Yim, *Cogging torque ripple minimization via position based characterization*, Proceedings of Robotics: Science and Systems (Berkeley, USA), July 2014.

[91] Rosalind Pickard, Chris Graszyk, Steve Mann, Josh Wachman, Len Pickard, and Lee Campbell, *VisTex database*, Media Lab., MIT, Cambridge, Massachusetts (1995).

[92] William R. Provancher, Mark R. Cutkosky, Katherine J. Kuchenbecker, and Günter Niemeyer, *Contact location display for haptic perception of curvature and object motion*, The International Journal of Robotics Research **24** (2005), no. 9, 691–702.

[93] Christopher Richard, Mark R. Cutkosky, and Karon MacLean, *Friction identification for haptic display*, Proc. of the ASME Dynamic Systems and Control Division, vol. 67, 1999, pp. 327–334.

[94] Robert Riener, Rainer Burgkart, Martin Frey, Bundit Sae-Kee, and Thomas Proll, *Force-torque input enhances medical vr applications*, Medicine Meets Virtual Reality **94** (2003), 282.

[95] Joseph M. Romano and Katherine J. Kuchenbecker, *Creating realistic virtual textures from contact acceleration data*, IEEE Transactions on Haptics **5** (2012), no. 2, 109–119.

[96] Joseph M. Romano, Takashi Yoshioka, and Katherine J. Kuchenbecker, *Automatic filter design for synthesis of haptic textures from recorded acceleration data*, Proc. IEEE International Conference on Robotics and Automation, May 2010, pp. 1815–1821.

[97] Satoshi Saga and Ramesh Raskar, *Simultaneous geometry and texture display based on lateral force for touchscreen*, Proc. SIGGRAPH Asia, ACM, 2012, p. 8.

[98] Mark Salada, Peter Vishton, J. Edward Colgate, and Eviatar Frankel, *Two experiments on the perception of slip at the fingertip*, Proc. IEEE Haptics Conference, IEEE, 2004, pp. 146–153.

[99] Guy Smith and Ian Burns, *MeasTex database*, 1997.

[100] Salvador Soto-Faraco and Gustavo Deco, *Multisensory contributions to the perception of vibrotactile events*, Behavioural Brain Research **196** (2009), no. 2, 145–154.

[101] Mandayam A Srinivasan and Cagatay Basdogan, *Haptics in virtual environments: Taxonomy, research status, and challenges*, Computers & Graphics **21** (1997), no. 4, 393–404.

[102] Yuta Takeuchi, Sho Kamuro, Kouta Minamizawa, and Susumu Tachi, *Haptic duplicator*, Proc. ACM Virtual Reality International Conference, 2012.

[103] Nikolaos G. Tsagarakis, T. Horne, and Darwin G. Caldwell, *Slip aestheasis: A portable 2D slip/skin stretch display for the fingertip*, Proc. IEEE World Haptics 2005, IEEE, 2005, pp. 214–219.

[104] Bertram Unger, Roberta Klatzky, and Ralph Hollis, *The physical basis of perceived roughness in virtual sinusoidal textures*, IEEE Transactions on Haptics **6** (2013), no. 4, 496–505.

[105] James Vallino and Christopher Brown, *Haptics in augmented reality*, Proc. IEEE International Conference on Multimedia Computing and Systems, vol. 1, IEEE, 1999, pp. 195–200.

[106] D. W. F. Van Krevelen and R Poelman, *A survey of augmented reality technologies, applications and limitations*, International Journal of Virtual Reality **9** (2010), no. 2, 1.

[107] Hari Vasudevan and M. Manivannan, *Recordable haptic textures*, Proc. IEEE International Workshop on Haptic Audio Visual Environments, 2006, pp. 130–133.

[108] Yon Visell and Shogo Okamoto, *Vibrotactile sensation and softness perception*, Multisensory Softness (Massimiliano Di Luca, ed.), Springer, 2014, pp. 31–47.

[109] Christopher R. Wagner, Susan J. Lederman, and Robert D. Howe, *Design and performance of a tactile shape display using rc servomotors*, Haptics-e **3** (2004), no. 4.

[110] Allan Weber, *The USC-SIPI image database*, Signal and Image Processing Institute of the University of Southern California. URL: http://sipi.usc.edu/services/database (1997).

[111] Robert J. Webster III, Todd E. Murphy, Lawton N. Verner, and Allison M. Okamura, *A novel two-dimensional tactile slip display: design, kinematics and perceptual experiments*, ACM Transactions on Applied Perception (TAP) **2** (2005), no. 2, 150–165.

[112] Parris Wellman and Robert D. Howe, *Towards realistic vibrotactile display in virtual environments*, Proc. of the ASME Dynamic Systems and Control Division **57** (1995), no. 2, 713–718.

[113] Michael Wiertlewski, Jose Lozada, and Vincent Hayward, *The spatial spectrum of tangential skin displacement can encode tactual texture*, IEEE Transactions on Robotics **27** (2011), 461–472.

[114] Götz Wintergerst, Ron Jagodzinski, Fabian Hemmert, Alexander Müller, and Gesche Joost, *Reflective haptics: enhancing stylus-based interactions on touch screens*, Haptics: Generating and Perceiving Tangible Sensations, Springer, 2010, pp. 360–366.

[115] K. Worden, C. X. Wong, U. Parlitz, A. Hornstein, D. Engster, Tegoeh Tjahjowidodo, Farid Al-Bender, D. D. Rizos, and S. D. Fassois, *Identification of pre-sliding and sliding friction dynamics: Grey box and black-box models*, Mechanical Systems and Signal Processing **21** (2007), no. 1, 514–534.

[116] Jeffrey M. Yau, Jonathon B. Olenczak, John F. Dammann, and Sliman J. Bensmaia, *Temporal frequency channels are linked across audition and touch*, Current Biology **19** (2009), no. 7, 561–566.

[117] Guangqi Ye, Jason J. Corso, Gregory D. Hager, and Allison M. Okamura, *VisHap: Augmented reality combining haptics and vision*, Proc. of IEEE International Conference on Systems, Man and Cybernetics, vol. 4, IEEE, 2003, pp. 3425–3431.

[118] Shunsuke Yoshimoto, Yoshihiro Kuroda, Yuki Uranishi, Masataka Imura, and Osamu Oshiro, *Roughness modulation of real materials using electrotactile augmentation*, Haptics: Neuroscience, Devices, Modeling, and Applications, Springer, 2014, pp. 10–17.

[119] Takashi Yoshioka, Sliman J. Bensmaia, Jim C. Craig, and Steven S. Hsiao, *Texture perception through direct and indirect touch: An analysis of perceptual space for tactile textures in two modes of exploration*, Somatosensory and Motor Research **24** (2007), no. 1-2, 53–70.

[120] Craig B. Zilles and J. Kenneth Salisbury, *A constraint-based god-object method for haptic display*, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, August 1995, pp. 146–151.