Doctoral Dissertation

Haptic Rendering of Surface 3D Curvature and Texture on Electrostatic Friction Display

Reza Haghighi Osgouei (하기기 오스고위 레자)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2017

정전기 마찰 디스플레이에서의 3차원 곡면 및 표면 햅틱 질감 렌더링

Haptic Rendering of Surface 3D Curvature and Texture on Electrostatic Friction Display

Rendering Surface 3D Curvature and Texture on Electrostatic Friction Display

by

Reza Haghighi Osgouei
Department of Computer Science and Engineering
Pohang University of Science and Technology

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

Pohang, Korea
12. 19. 2017
Approved by
Seungmoon Choi (Signature) Seungmoon Choi
Academic advisor

Rendering Surface 3D Curvature and Texture on Electrostatic Friction Display

Reza Haghighi Osgouei

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH

12. 19. 2017

Committee Chair	Seungmoon Choi	(Seal)
Member	Seungyong Lee	(Seal)
Member	Hee-Kap Ahn	(Seal)
Member	Seungjin Choi	(Seal)
Member	Soohee Han	(Seal)

DCSE 20131172

하기기 오스고위 레자. Reza Haghighi Osgouei
Haptic Rendering of Surface 3D Curvature and Texture on Electrostatic Friction Display,
정전기 마찰 디스플레이에서의 3차원 곡면 및 표면 햅틱 질감 렌더링
Department of Computer Science and Engineering , 2017, 121p, Advisor : Seungmoon Choi. Text in English.

ABSTRACT

In this work, we address the problem of rendering surface curvature and fine texture using an electrovibration display. We proposed effective algorithms to address each problem separately. In the first part, we introduced a gradient based method to render 3D objects on an electrovibration display. It includes a generalized real-time algorithm to estimate surface gradient from the surface of any 3D mesh. In addition, a separate edge detection method is included to emphasis sharp edges while scanning the surface of a mesh. Conducting a human user study, we showed that in the presence of haptic feedback generated using our algorithm, the users can better recognize 3D bumps and holes when the visual feedback is limited and puzzling.

In the second part, we proposed a neural network based texture modeling and rendering method. We first created an inverse neural network dynamic model for the electrovibration display to estimate an actuation signal from the forces collected from the surface of real texture samples. For the force measurement, we developed a linear motorized tribometer enabling adjusting applied normal pressure and scanning velocity. We trained neural networks to learn from the forces resulted from applying a full-band PRBS (pseudo-random binary signal) to the electrovibration display and generate similar actuation signals. While the networks are trained under known normal pressure and scanning velocity, for untested conditions, we proposed a two-part interpolation scheme to produce actuation signal from the neighborhood conditions. The first part, generates a signal by taking weighted average between the signals with the same scanning velocity but different masses. The second part, performs a re-sampling process, either down-sampling or up-sampling, on the newly estimated signals to produce a final signal according to the user applied normal pressure and scanning velocity. We conducted a user study to put the proposed algorithm on test. We asked users to rate the similarity between a real texture and its virtual counterpart. The experimental setup included a load-cell to measure user applied pressure and an IR-frame to track his/her finger position and eventually calculate user's scanning velocity. Testing six different real texture samples, we achieved an average similarity score of 60% using the proposed algorithm against 39% using a basic record-and-playback method. This revealed the potentials of the proposed texture modeling and rendering algorithm accompanied by a linear interpolation scheme in creating virtual textures similar to the real ones.

Contents

I. Introduction **1**

1.1 Research Overview 3

1.2 Contributions 5

II. Related Work **7**

2.1 Electrostatic Friction Display 7

2.2 Rendering Surface Geometry 11

2.3 Rendering Surface Texture 13

III. Basic Gradient Algorithm For Rendering 3D Geometries **21**

3.1 Device Characterization 22

3.1.1 Tribometer 22

3.1.2 Output Characteristics 24

3.2 Formative user study 27

3.2.1 Force Profiles 29

3.2.2 Participants 34

3.2.3 Procedure 35

3.3 Results and Discussion 36

3.3.1 Open Descriptions 36

3.3.2 Closed Selections 37

3.3.3 Summary and Discussion 40

3.4 Conclusions 41

IV. Generalized Gradient Algorithm For Rendering 3D Meshes 43

4.1 Fundamental Algorithm 43

4.1.1 Computing Minimum and Maximum Force 46

4.1.2 Edge Emphasis 47

4.2 Summative User Study 49

4.2.1 Methods 49

4.2.2 Results 53

4.2.3 Discussion 56

4.3 Conclusions 57

V. Data-driven Texture Rendering 61

5.1 Data Collection 62

5.1.1 Tribometer 62

5.1.2 Electrovibration Display 64

5.2 Texture Model 65

5.2.1 Inverse Dynamics Model 65

5.2.2 Synthesizing Actuation Signals 69

5.2.3 Error Metric 71

5.3 Experimental Results 71

5.3.1 Baseline Performance 73

5.3.2 Training Performance 74

5.3.3 Cross-validation 76

5.4 Conclusions 78

VI. Improved Interpolation Scheme **80**

6.1 Problem definition 81

6.2 Potential solution 82

6.3 Test on real texture samples 86

6.4 Improved interpolation scheme 90

6.5 Human user study 92

  6.5.1 Methods 93

  6.5.2 Results 97

  6.5.3 Discussion 103

6.6 Conclusion 105

VII. Conclusions **108**

References **111**

List of Tables

3.1 Experimental conditions for force-feedback device. 31

3.2 Experimental conditions for electrostatic device. 31

4.1 Results of Four-way Within-Subject ANOVA. 54

5.1 Adjusted gains for each material (TS1–TS6) under each experimental condition. 75

6.1 The mean and standard deviation values of the similarity score between real textures. 98

6.2 The mean and standard deviation values of the similarity score between real and virtual textures. The values inside parenthesis are the ratios of the above values to the upper bound (89%). . . . 100

6.3 The mean and standard deviation values of the lower and upper bounds. 100

6.4 The mean and standard deviation values of minimum and maximum scores between real and virtual textures. 101

6.5 The mean and standard deviation values of the similarity score between real and virtual textures comparing nn-based and rp-based methods. 103

List of Figures

2.1 The original illustration from [1]. Left: Gradient Technique. Right: detail of local spring force computation for x direction. 12

3.1 Rotary tribometer. 23

3.2 Graphical user interface for device characterization. When each vertical line is crossed, a haptic grain is rendered. 25

3.3 Example of force measurements. Blue: tangential force and red: normal force. (a) Raw data. (b) Detail from the region highlighted with two vertical lines in (a). (c) Peak-to-peak amplitude of tangential force vs. input intensity. 26

3.4 Static behavior of the electrostatic friction display. (a) Raw tangential force data. (b) After filtering, the data were averaged to estimate the static friction force increase. (c) Static friction force increase vs. input intensity. 28

3.5 Gaussian bump (blue) and the corresponding force profile (red) taken from [2]. The scanning direction is from left to right. 29

3.6 Profiles for the electrostatic display. Blue: geometry [cm], red: force [N], and orange: intensity. 32

3.7 Measured friction force profiles for each experimental condition. Blue: measured force profile [N] (only absolute values are shown for clarity), red: geometry profile, orange: intensity profile (scaled to show the trend). 33

3.8 Correct recognition ratios using the PHANToM (mean 91%) and the Feelscreen tablet (mean 64%). Error bars show standard errors. 38

3.9 Correct recognition ratios with the PHANToM for each experimental condition (see Table 3.1). 39

3.10 Correct recognition ratios with the Feelscreen tablet for each experimental condition (see Table 3.2). 39

4.1 Variables for the generalized lateral force rendering algorithm. . . 44

4.2 Dihedral angles. Thick lines represent the cross-section of polygons. 48

4.3 Plots of haptic grain AREA-EVEN (a) and EDGE-TICK (b), and their spectrums (c and d). AREA-EVEN has a single main spectral component, and this effect generates smooth electrostatic force. EDGE-TICK includes multiple spectral components, which makes it feel rough and bumpy [3]. EDGE-TICK conveys an impression of sudden impact and is suitable for rendering sharp edges. 50

4.4 (a) 3D mesh of a Monkey model. (b) Results of edge emphasis. All convex edges are highlighted in blue. Among them, those with $\theta < 135°$ (relatively sharp ones) are marked in red. 51

4.5 3D shapes used in the summative user study. (a) Gaussian bump: width 6, length 6, height 1.8, $z(x, y) = \frac{8}{1.4\pi}\exp(\frac{-x^2-y^2}{1.4})$. (b) Square frustum bump: width 6, length 6, height 2, base square 3×3, top square 1×1. All dimensional units are cm. 51

4.6 Experimental conditions. Twenty scenes were composed by combining two shapes (Gaussian and square frustum), two lighting conditions (spotlight and directional), and five configurations (0B4H, 1B3H, 2B2H, 3B1H, and 4B0H). From left to right: Gaussian-spotlight, Gaussian-directional light, square frustum-spotlight, and square frustum-directional light. From top to bottom: 0B4H, 1B3H, 2B2H, 3B1H, and 4B0H. Some scenes can be easily confused because the direction of directional light in not known to users. For instance, the (2, 2) scene (Gaussian-spotlight, 1B3H) can be mistaken with the (4, 2) scene (Gaussian-spotlight, 3B1H). 58

4.7 Snapshot of the GUI program. In the lower right corner, a button was provided to move to the next scene after providing answers using the four toggle buttons in the lower left corner. Participants checked each toggle button for bump and unchecked it for hole, for the corresponding object in the scene. 59

4.8 Correct recognition ratios for task (V 75% and V+H 95%) and light (spotlight 81% and directional 88%). Error bars show 95% confidence intervals. 60

4.9 Interaction effects between task and light. Error bars show 95% confidence intervals. 60

5.1 Motorized linear tribometer. The measurement stylus is attached to the moving platform using an Aluminum link. 63
5.2 An example PRBS9 with $n = 9$. 67
5.3 Block diagram of the training and evaluation of NARX network. $y[n]$ is the reference signal (PRBS9) and $x[n]$ is the external input (recorded lateral force). 69
5.4 Example result of the training and evaluation of NARX network. For clarity, only 300 ms of the recordings are shown. The blue solid line is the reference PRBS9, and the dashed red line is an estimate from the neural network. 70
5.5 Six texture samples used in the evaluation. From left to right: dotted sheet (TS1), chair fabric (TS2), felt fabric (TS3), painting canvas (TS4), transparent plastic sheet (TS5), and scrunched paper (TS6). 72
5.6 Force-velocity grid for cross-validation. m_i indicates mass and v_j scanning velocity. For experimental conditions highlighted by blue squares, the actuation signal is obtained from the corresponding trained neural networks. For others marked by red circles, the signals are obtained by linearly interpolating neighbors. 72

5.7 Baseline FFT plots with error metrics. Dotted black line is the first repetition, and solid green line is the second repetition. The average errors are given inside parentheses on the top and left side. 73

5.8 FFT plots comparing virtual and real textural forces. Dotted black lines indicate real forces, and solid green lines virtual ones. Average errors are given inside the parentheses on the top and left sides. . . 77

5.9 FFT plots for cross-validation comparing virtual and real textural forces for the experimental conditions not used for neural network training. Dotted black lines indicate real forces, and solid green lines virtual ones. Average errors are given inside the parentheses on the top and left sides. 78

6.1 Example of two sine waves with frequencies $f_1 = 10$ Hz and $f_2 = 20$ Hz, and their corresponding FFT plots. $y_3 = 0.5y_1 + 0.5y_2$ is calculated as an estimation for $f_3 = 15$ Hz by taking weighted average from y_1 and y_2. Its time and frequency domain plots are given on the right. 83

6.2 Example of re-sampling two sine waves with frequencies $f_1 = 10$ Hz and $f_2 = 20$ Hz to generate a signal (sine wave) at the desired frequency of $f_3 = 15$ Hz. Time domain plots are given on the top and FFT plots on the bottom. The red plots are the original signals and the blue ones are re-sampled ones. y_1^r, blue plot, is the re-sampled version of y_1 and y_2^r is of y_2. y_3 is obtained by linearly combining y_1^r and y_2^r: $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 15) = 0.5y_1^r + 0.5y_2^r$. . 85

6.3 Another example of re-sampling and combining two sine waves with different frequencies and different amplitudes. For the desired frequency of $f_3 = 19$ Hz the signal is obtained from $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 19) = 0.1y_1^r + 0.9y_2^r$. 86

6.4 Another example of re-sampling and combining two sine waves with different frequencies and different amplitudes. For the desired frequency of $f_3 = 12$ Hz the signal is obtained from $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 12) = 0.8y_1^r + 0.2y_2^r$. 87

6.5 Rippled paper with almost perfect sinusoidal surface patterns. . . . 88

6.6 Power spectra of forces collected from the surface of rippled paper under five different scanning velocities. The inset plot at the upper right corner, shows a linear relationship between the frequencies of the main component and their corresponding velocities. 89

6.7 Improved interpolation scheme. (m_u, v_u) denotes user applied pressure and velocity. Blue squares are the adjacent actuation signals obtained from neural networks trained under corresponding experimental conditions. Blue arrows indicate sup-steps 1.1 and 1.2, applied on the vertical nodes to yield blue dashed circles. Long green arrows indicate sub-steps 2.1 and 2.2, applied on the horizontal nodes to yield green dashed circles. Short green arrows, indicate the final sub-step 2.3, applied on the re-sampled signals (green dashed circles) to obtain green square, the output of interpolation. 92

6.8 Real vs virtual spectra for rippled paper. The cross-validation conditions include one normal pressure (85 g) and four scanning velocities (3.5, 4.5, 5.5, and 6.5 cm/s). Recordings from rippled paper are shown with dotted black lines and recordings from 3M panel actuated with the interpolated signals with solid green lines. In all four conditions, a satisfactory agreement between the two spectra, real vs. virtual, can be seen, proving the effectiveness of the proposed interpolation scheme. The error metric values, E_s, are shown on top of each plot. 93

6.9 Experimental setup used for human user study. 3M capacitive touchscreen is used as electrovibration display. Under the panel a load cell is installed to measure user's pressure and on top of it an IR-frame to track user's finger position. 95

6.10 Average similarity scores comparing two real textures. Left: two different real materials. The red bars are for highly-different group and the blue bars for moderately-different group. Right: two similar real materials. Error bars show standard errors. 98

6.11 Average similarity scores comparing real and virtual textures. Error bars indicate standard errors. The yellow bars show lower and upper bounds, with lower(3) including only highly-different group (3 pairs) and lower(6) both highly-different and moderately-different groups (6 pairs). 99

6.12 Average minimum and maximum scores for each material. 101

6.13 Comparison between neural network-based (nn) and record-and-playback (rp) methods. Error bars indicate standard errors. 102

Chapter I.
Introduction

Touch is the most fundamental sense we are equipped with from the moment we enter this world. Even new born babies know how to use their sense of touch to interact with their surroundings. Touch is a necessity in doing ordinary tasks around us which without it even a very simple task would be challenging to complete. Just imagine how difficult it can be to pick an object if you cannot feel its shape or determine the amount of force you need to apply to hold it. Touch is very important to us in many levels and we rely on our touch sense more than we think we do.

Modern technologies in this digital era added new interactive agents around us which require our touch input. Touchscreen consumer electronics such as smart phones and tablet devices are among them. They are a versatile device that displays visual content and takes touch input simultaneously. More specifically, smart phones are an inevitable part of our daily life. Users spend a significant amount of time interacting with the digital contents on their mobile phones. So equipping such devices with a functionality to provide some sort of touch feedback was inevitable and seemed to be a natural course of technological development. However, despite the technological advances, these devices lack the ability of providing programmed tactile feedback, which can be essential for more natural and intuitive interaction. At best, they provide some simple monotonic vibration patterns in response to the user's touch input. This is neither appealing nor satisfactory given the expectations users have from such modern devices.

With the introduction of variable friction displays, this limitation has been addressed by technologies collectively called *surface haptics*. These technologies modulate friction between a user's fingertip and a touchscreen surface in order to create a variety of tactile sensations when the finger explores on the touchscreen. This functionality allows the user to see and feel the digital content simultaneously with richer haptic information, leading to improved user experience and/or usability. There exist two major approaches in surface haptics: electrovibration and ultrasonic vibration. Whereas the former increases the surface friction by modulating attractive electrostatic force, the latter decreases the friction by vibrating the surface at an ultrasonic frequency and creating an air gap. Such electrovibration displays have the advantages that they require only electrical components and that the friction can be controlled uniformly on the screen, which are particularly attractive for mobile devices with a provision of adequate amplifiers.

The lack of appropriate tactile feedback motivated us to seek effective ways to tackle the challenge employing electrovibration technology. More specifically, in this work, we focus on rendering surface features such as geometry and texture on an electrovibration display. Many applications can benefit from this added functionality, such as Internet shopping, education, security authentication, entertainment, and etc. We believe in close future mobile phones will be equipped with such a functionality and hence algorithms such as ours will be an asset taking full advantage of this added quality.

1.1 Research Overview

The focus of this research is twofold: rendering surface 3D geometries and surface fine texture on an electrovibration display. We address each fold in separate chapters.

In chapter 2, we review technological background of the electrovibration and the early displays fabricated based on it. We then review the recent displays, including the one introduced in TeslaTouch project [4]. Later on, we separately survey the literature regarding the work related to rendering 3D features and surface textural patterns. The focus for the latter on is on data-driven texture modeling and rendering methods.

The focus of chapter 3 is on rendering 3D geometries and seeking an effective way to improve the 3D shape perception of visual objects displayed on a touchscreen by providing electrovibration feedback We first present the physical characteristics of the electrostatic friction display used in our research. This is followed by a formative user study that investigated whether users can identify

3D features, such as bumps and holes, from electrovibration feedback alone without any visual information. We adapted the basic gradient-based lateral force rendering algorithm studied in [2] for that. The results of the formative user study provided useful information as to how to utilize frictional force feedback for our purpose.

Based on this finding, in chapter 4, we extended and generalized the basic algorithm to estimate the surface gradient for any given 3D mesh. Also, we proposed an additional edge emphasis algorithm for rendering sharp edges. Finally, we conducted a summative user study to quantify the performance improvement of 3D object shape recognition enable by our algorithm under limited visual conditions. To our knowledge, this work is among the first for 3D shape rendering using electrostatic friction displays.

The focus of chapter 5 is on data-driven texture modeling and rendering. It includes recording texture data from the real samples, building texture models and rendering them on an electrovibration display. We investigate the application of NARX neural networks on capturing the dynamic behavior of the electrovibration display by inversely learning from the forces resulted from applying a full-band pseudo-random binary signal (PRBS) to the display. We show how to synthesize actuation signals using the trained neural networks under known experimental conditions based on the forces collected from the surface of real materials. We also introduce a basic interpolation scheme to estimate actuation signals for untested experimental conditions from adjacent nodes in the pressure-velocity grid.

Observing the poor performance of the basic interpolation scheme estimating the signals for the conditions with different scanning velocities, the aim of chapter 6 is to find an effective solution. We pay a closer look into the problem and propose a potential solution based on re-sampling neighboring signals. This yields to an improved two-part interpolation scheme. In the first part, we take weighted average between the adjacent signals having the same scanning velocity but different masses. The lower and higher normal pressures (masses) and the user applied pressure define the weights. In the second part, we apply a re-sampling process, either down-sampling or up-sampling, on the newly produced signals and then take a weighted average with the lower and higher velocities and the user applied velocity as the weights. At the end we conduct a human user study to evaluate the performance of the proposed texture modeling and rendering method on creating virtual textures similar to the real ones.

The manuscript ends with the conclusions and future remarks.

1.2 Contributions

The major contributions of this study are summarized bellow:

- Of first to consider lateral forces as the salient surface texture feature
- First to use a touch pen (instead of human finger-tip) to capture lateral forces from the surface of texture samples using a motorized tribometer
- Developed a generalized real-time algorithm to estimate surface local gradient at the touch point from any 3D mesh

- Devised a neural-network-based inverse model describing the dynamic behavior of the friction display with respect to a full-band pseudo-binary random signal (PRBS) as actuation signal
- Proposed a two-part interpolation scheme to estimate an actuation signal for an untested condition from the adjacent tested conditions

Chapter II.
Related Work

2.1 Electrostatic Friction Display

Mallinckrodt et al. in 1953 was the first to discover that touching an insulated metalic surface, connected to a 110 V power line, creates a tingling rubber-like feeling on the electrically grounded finger [5]. This phenomena is called *electrovibration* by Grimnes in 1983, explaining its principle of operation based on Coulomb's electrostatic force [6]. Electrovibration is due to the electrostatic attraction force between two conductive plates separated by a dielectric. When the finger scans an insulated electrode, a condenser is formed between the electrode and the conductive substance under the skin. Exciting the electrode using a periodic voltage induces electrostatic attraction, and this increases the friction force between the surface and the moving finger. This electrostatic stimulation was introduced into a tactile display by Strong et al. [7]. They developed the first electrostatic display using a stimulator array consisting of a large number of small electrodes. They reported that the intensity of the perceived vibration was mainly due to the peak applied voltage. Later, a polyimide-on-silicon electrostatic fingertip tactile display was fabricated with 49 electrodes arranged in a square array [8]. They conducted experiments to assess the intensity and spatial resolution of the tactile percepts. In a following study, its application to present various spatial tactile patterns such has line, triangle, square, and circle to the visually impaired users is investigated [9]. In all these work, the dryness of fingertip is emphasized to be the key factor maintaining the percept, reporting that a small amount of sweat could cause the percept to fade or disappear. Indirect stimulation was suggested as a solution. Yamamoto et al. built a display with a thin slider film between electrostatic stator electrodes and fingertip for presenting surface roughness [10]. In another work, multiple contact pads are used for multi-finger interaction with a large electrostatic display [11].

Despite these early work, TeslaTouch, developed at Disney Research, was the first to adopt electrovibration technology as an effective haptic interface for touchscreens [4]. The core of TeslaTouch is a transparent capacitive touch panel (Microtouch, 3M, USA [12]) driven by a high voltage signal to modulate friction on a sliding finger. The panel is made of a thick glass layer on the bottom, a transparent electrode (indium tin oxide; ITO) in the middle, and a thin insulator layer on the top. In the usual setup, the electrode is excited by high AC voltage, and the human body is grounded electrically. The big advantage of TeslaTouch is that the capacitive panel is a commercial off-the-shelf product which requires only an additional high voltage amplifier for proper operation. The same panel has been used in electrovibration displays by other groups [13–19]. Radivojevic et al. at Nokia, introduced a flexible and bendable version by replacing indium-thin oxide (ITO) with Graphene [20]. Another company in Finland, Senseg, developed Tixel [21], a transparent electrostatic film targeting hand-held devices. Senseg later introduced a short-lived commercial product called Feelscreen, a 7" Android tablet overlaid with Tixel, into the market between 2014 and 2016. Feelscreen has been used in several projects such as 3D shape rendering [22], texture gradients [23], and visual and haptic latency [24]. At the moment, Tanvas [25], a startup company in USA, is commercializing similar products but on a larger 10" tablet with some improvements such as generating stronger friction forces and not requiring an external power supply.

Some other researchers developed their own electrovibration display not using the 3M capacitive touch panel. Pyo et al. built a tactile display that provides both electrovibration and mechanical vibration on a large surface [26]. They fabricated an insulated ITO electrode on top of an electrostatic parallel plate actuator, both operating based on the electrostatic principle. A non-transparent electrostatic friction display was also developed in [27, 28] using an aluminum plate covered with a thin plastic insulator film.

These displays do not support multi-touch or localized friction modulation, and all fingers in contact with the surface experience the same sensation. This issue was addressed by several prototypes presenting local stimulation. For example, a display panel was developed with multiple horizontal and vertical ITO electrodes in a grid enabling localized stimulation at the region where the vertical and horizontal electrodes cross each other [29]. In [11], a multi-finger electrostatic display was developed consisting of a transparent electrode and multiple contact pads on which users place their fingers. Applying different voltages to the pads and electrically grounding the transparent electrode induce different frictional stimuli to the multiple fingers.

As well as fabrication, various properties of electrovibration have been investigated too. The polarity effect of the actuation signal is studied in [30], reporting that tactile sensation is more sensitive to negative than positive pulses. Meyer et al. showed an expected square law dependence of frictional force, measured by a tribometer, on actuation voltage [13]. A similar approach is taken by Vezzoli et al. to develop a model for electrovibration effect considering frequency dependence [28]. Kim et al. proposed a current control method to provide more uniform perceived intensity of electrovibration [16]. In another work and by comparing two actuation signals, it is reported that square waves are more detectable than sine waves at frequencies lower than 60 Hz while they are same at higher frequencies [31]. Testing three methods, amplitude modulation, adding dc offset, and their combination, Kang et al. investigated low voltage operation of electrovibration display [19]. They showed all methods increased dynamic friction force while only dc offset increased static friction force.

The relationship between input signal and output friction in electrostatic friction displays is not clearly understood and a number of studies have shown great interest in defining such relationship. Researchers have worked on this topic either by measuring friction forces using a tribometer [13, 28] or by estimating perceived intensities in psychophysical experiments [14, 32]. For instance, Meyer et al. [13] developed a tribometer to make precise measurements of finger friction and confirmed the expected square law of frictional force to driving voltage. They also showed a linear mapping between friction and normal force, confirming the Coulombic model of dry friction. Conducting a 6-value effect strength subjective index rating, Wijekoon et al. showed a significant correlation (0.8) between signal amplitude and perceived intensity but no correlation between frequency and perceived intensity [32]. In [14], participants assigned a number between 0 and 100 to the subjective friction intensity. A linear fit in log-log scale was observed in the normalized results relating applied voltage amplitude to perceived friction force intensity.

2.2 Rendering Surface Geometry

Rendering 3D objects on a flat surface, either using a haptic interface or a variable friction display, has not been addressed much in the literature. In an early work regarding haptic perception of curvature, Gordon and Morison showed that the gradient is an effective stimulus for curvature perception and humans rely on local curvature when perceiving surface [33]. Later, Minsky et al. demonstarted that tangential force alone can be sufficient for rendering surface texture assuming it is made of little bumps [1]. They introduced *Gradient Technique* to create the illusion of bumps and valleys using a 2D force-feedback joystick. As the user moves the joystick in a direction which is *up* a bump, his motion is opposed by a spring force proportional to the height of the bump. This gives the sense that it is very difficult to move to the top of the bump (springs resist being stretched), and easy to fall off the bump back into a lower region of the simulated surface (springs like to revert to a short length). For fine grained surfaces joystick spring forces can be computed based on a local gradient. As the user moves the joystick on the virtual surface, the change in height in the direction of motion is noted . We create virtual springs opposing the motion "up" the sides of each tiny bump. Thus the spring forces applied to the hand are computed from local gradients of the height of the surface (Fig. 2.1).

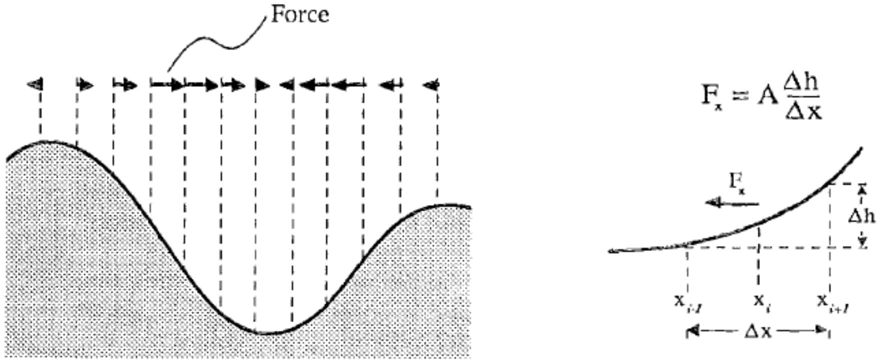


Figure 2.1: The original illustration from [1]. Left: Gradient Technique. Right: detail of local spring force computation for x direction.

Based on the *Gradient Technique*, an early attempt to create the haptic illusion of a non-flat shape on a nominally flat surface was introduced in [34] using a force-shading algorithm. Later continuing their earlier work [35], Robles-De-La-Torre and Hayward demonstrated that in active exploration of a physical shape, lateral force applied to the sliding finger plays the main role in the perception of shape [2]. They investigated the accuracy of physical shape recognition using a one-DoF (degree of freedom) force-feedback device without visual cues. Different combinations of physical and virtual geometries (bump, hole, and flat surface), e.g., a virtual bump laid on a physical flat surface, were presented to participants. The virtual shapes were rendered using lateral force only. Participants could accurately identify the virtual shapes in all conditions.

This study was foundational to the gradient-based algorithm of Kim et al. [14] for rendering 3D features on a touchscreen using electrovibration. In their work, a psychophysical perceptual model, subjectively relating the perceived friction to the applied voltage, was formulated. The model was a straight-line in log-log scale, fitted over averaged users' ratings of the perceived friction intensity in a scale of 0–100. The model then utilized to modulate friction and render three lateral force profiles: height, slope and rectangular profiles. They compared users' preference for three types of force profile for a visual bump displayed on the screen. Results indicated that the slope profile best matched the visual bump. They generalized this finding to a 2D gradient-based rendering algorithm for 3D features and applied the algorithm to many user interface examples.

2.3 Rendering Surface Texture

Haptic texture rendering has been always a challenge in the haptics community. It is all about creating virtual textures that perfectly mimic the feel of specific real textures. While including textures has the potential to increase the realism of haptic virtual environments, but they must be implemented in a way that respects the software and hardware capabilities of haptic interface systems. Researchers have thus developed several different approaches to haptic texture rendering [36–40]. The focus of this work is a data-driven texture modeling and rendering method using an electrovibration display. Therefore, we limit the scope of this survey to the work either employing a data-driven method or targeting a flat display.

Data-driven, or measurement-based haptic rendering, is a general approach that uses recordings from real objects to generate realistic haptic feedback in virtual environments [41, 42]. It can be either parametric and physics-based, to optimize parameters of a predefined model, or non-parametric and generic. It is usually accompanied by a generic interpolation scheme to handle the data sets not being measured. This data-driven approach enables researchers to bypass the complex step of hand tuning a dynamic simulation of the target interaction to try to match a haptic sensation. Instead, the goal of the modeling process is to capture the output response of the system (e.g., force and acceleration) given some set of user inputs (e.g., position, velocity, and force). Such methods shift the focus from reproducing the physics of the interaction to reproducing the real sensations felt by the user, and thus they have been largely successful at realistic haptic simulation [43].

Manual surface exploration using a handheld sensorized stylus has been the prevalent data collection method for isotropic texture modeling. They are inexpensive and easy to use, and also allow for free-form surface scanning. For example, Pai and Rizun developed a wireless haptic texture sensor (WHaT) equipped with a 3D accelerometer and a 1D force sensor, all packaged compactly in a stylus [44].

Using WHaT along with a visual marker for 3D position tracking, Lang and his colleagues presented a series of studies for data-driven haptic modeling and rendering in virtual environments [45–47]. Their framework is mainly for compliance and texture. The compliance is dynamically estimated from a linear relationship between user-applied force and resulting acceleration. For texture modeling, a height profile is built from acceleration data using amplitude scaling (to remove the effect of applied force) followed by Velvet integration. This height profile is registered onto the object surface along the scan path for rendering with a force-feedback interface. An alternate method proposed by the same group makes an acceleration model [48, 49]. Scanned acceleration data is automatically segmented so that each segment includes a single decaying vibration. Each decaying vibration is modeled using an infinite impulse response (IIR) filter. These IIR filters are registered onto the object mesh following the scan path. Vibration amplitude is scaled using a linear function of scan velocity and applied force.

Another series of work has been contacted by Kucknbecker's group for data-driven texture rendering based on contact accelerations. In their early work [43], they collected contact accelerations under constrained conditions using a rotating drum with texture samples attached to its exterior surface and a hand-held prob equipped with three-axis accelerometer. The data from four different textures are recorded under fifteen different velocities and eight different force levels, resulting in a total number of 480 data sets. An autoregressive (AR) model based on linear predictive coding (LPC) is fit to each data set. The model is optimized by minimizing the residual error between recorded and predicted accelerations. The model resembles an IIR filter of order n. Once trained, the inverse model is used to synthesize accelerations by feeding a Gaussian white noise with zero mean and variance equal to average signal power remaining in residual. For scanning velocities and normal forces not used in training, acceleration values are calculated by a model for which its coefficients are bilinearly interpolated from four adjacent models. Later, they improved their stylus by adding a force sensor and two voice-coil actuators to the three-axis accelerometer for haptic interaction with a tablet computer [50]. The same stylus is used for both data collection and texture rendering. Using the same modeling and synthesis methods and conducting a user study, they reported moderately high similarity ratings (M 65.4, SD 19 in a 0–100 scale) between real and virtual textures. Their model was further refined by replacing LPC with an autoregressive moving average (ARMA) to better handle the weak stationary nature of the data and also to reduce the size of model [51]. In their next work [52], data collection are conducted manually under free conditions not constrained by predefined scanning velocities and normal forces. They applied Auto-PARM algorithm to automatically segment recorded data. With an AR model for each segment, this new approach resulted in many simpler AR models in the space of scanning velocity and normal force. They recently opened a public repository of one hundred acceleration-based texture models [53], along with associated modeling and rendering software including a method to render the texture models using a force-reflecting interface. This is a pivotal contribution for the advancement of haptic texture modeling and rendering. In their last work, they investigated the importance of matching between physical friction, hardness, and texture in creating realistic haptic virtual surfaces [54]. The virtual surfaces were created using a combination of friction, tapping transient, and texture vibration models to capture the full haptic experience, and they were rendered using a 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. Piecewise autoregressive texture models were created to represent the vibrations induced in a tool as it is dragged across a textured surface. A synthetic vibration signal is generated and displayed through a voice-coil actuator attached to the tip of the Omni. Conducting a user study, they reported that the realism improvement achieved by including friction, tapping, or texture in the rendering was directly related to the intensity of the surface's property in that domain (slipperiness, hardness, or roughness).

A special attention to anisotropic textures has been also given. Shin et al. compared texture modeling using unified and frequency decomposed neural networks with the former being capable of handling anisotropic patterns [55]. In addition, a dedicated data-segmentation and interpolation method based on

Radial Basis Function Network (RBFN) for anisotropic textures is proposed in [56].

Despite these endeavors using conventional or customized haptic interfaces, little work has been done on data-driven texture rendering using variable friction displays or particularity electrovibration displays.

An electrostatic friction display creates clearly perceptible stimuli when the surface is laterally scanned, but not when the finger is stationary. This fundamental limitation has confined the application of electrostatic friction displays mostly to texture rendering. In the only relevant work [15], Ilkhani et al. proposed a data-driven texture rendering method by recording accelerations from three real materials and playing them back on an electrovibration display. Their automated data collection is done under single constraint condition (contact force 0.35 N and scanning velocity 0.74 m/s) using a servomotor controlled by an Arduino Uno. They conducted a user study to compare the perceived surface roughness generated with their data-driven signals and with that of square wave signals. The frequency of each square wave is set based on the main frequency of the corresponding acceleration. Using a visual indicator, they made user to keep a constant scanning velocity, but not equal to the data-collection velocity and presumably very slower than that. In addition, there is no mention of contact force status during experimentation. Nevertheless, they reported higher percentage of similarity between data-driven textures and real ones in comparison with square wave patterns. In their extended work [57], they applied the same approach on the data from Penn Haptic Texture Toolkit [58] and performed MDS analysis to create a perceptual space and to extract underlying dimensions of the textures. Their results showed roughness and stickiness as the primary dimensions of texture perception.

In [59], authors presented a high-fidelity surface haptic display for texture rendering using a non-contact position sensor and a low-latency rendering scheme. The friction was controlled by modulating amplitude of ultrasonic vibrations of a glass plate while a finger was sliding along the plate. They applied a lead-lag compensator to correct the amplitude attenuation and a high-order filter to address the effects due to the frictional mechanics of the finger. To achieve a better flatness of the force frequency response and a better time-domain tracking performance, at the expense of a more complex implementation, high-order filters were utilized. The device can reproduce features as small as 25 μm while maintaining an update rate of 5 kHz. Signal attenuation, inherent to resonant devices, is compensated with a feedforward filter, enabling an artifact-free rendering of virtual textures on a glass plate.

In [60], an approach for texture simulation on a friction control device based on comparison of the finger position with a pre-compiled map of friction is proposed. Their device incorporated an open loop control of the friction on the finger sliding on an ultrasonic vibrating plate. The key challenge was the bandwidth of the position sensor determines the maximum reproduction bandwidth of the device which can be as law as 50 Hz for a capacitive touchscreen which is largely insufficient to reproduce real textures. Two different texture rendering schemes were introduced, a classic one for object representation (SHO: surface haptic object) and a digital synthesis for texture representation (SHT: surface haptic texture). By defining two different signals, the SHO, spatially located, and the SHT, spatially periodical, they proposed a solution to overcome the limitation of the slow position acquisition. They conducted a psychophysical experiment to analyze the advantages of the two texture rendering techniques concluding that the SHT approach performed significantly better than the SHO for a large spatial frequency (17 stimuli/cm) while it was the opposite for a spatial frequency around (0.7 stimuli/cm).

Chapter III.
Basic Gradient Algorithm For Rendering 3D Geometries

In this chapter, we investigate whether lateral forces can be used to render basic 3D geometries on an electrovibration display. The study was motivated by our on-going research for an integration of a multi-focus autostereoscopic 3D visual display and an electrostatic display onto a touchscreen. Multi-focus displays provide greatly superior 3D visual perception than regular touchscreens, and we have been seeking the methods to further enhance 3D perception by means of haptic feedback. The present study was carried out for the following two research questions:

Q1 *Can users identify primitive 3D features, such as bumps and holes, from electrovibration alone without any visualization?*

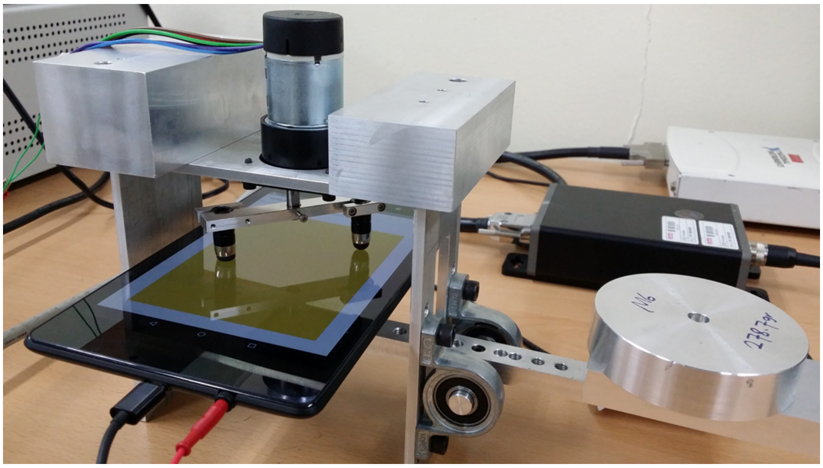
Q2 *How close is the recognition performance to that of the case using an active kinesthetic interface?*

3.1 Device Characterization

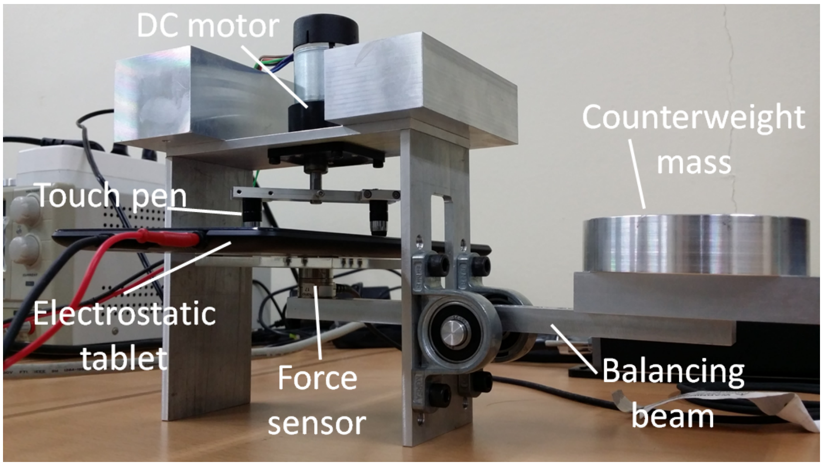
In this study, we used a Feelscreen development kit (Senseg, Finland) for a tablet, in which an electrostatic display was overlaid on a commercial tablet (Google Nexus 7). This device can provide strong and clear sensations of electrovibration. Its software development kit (SDK) supported nine haptic effects (called haptic grains) that resulted in noticeably different friction patterns. The intensity of haptic grain could be controlled using a normalized value (0.0–1.0). However, the Feelscreen SDK did not allow completely-customable input, e.g., a sinusoidal wave with certain frequency and amplitude, and the characteristics of generated friction forces was unknown. Therefore, it was necessary to characterize FeelScreen's various haptic grains, and we built a tribometer for that purpose.

3.1.1 Tribometer

Our tribometer is similar to those used in the research of electrostatic displays [13, 28], but also has a few differences. The previous studies required great care in controlling the electrical skin impedance of fingertip since it depends greatly on person, temperature, and moisture. In particular, the skin moisture level can change even in a short period of use. We found that the electrostatic display of Feelscreen also responded to some touch pens. The sensations of electrovibration resulted from the use of such a touch pen and the bare finger were very similar. Hence, our tribometer uses a touch pen instead of the human fingertip for data collection in order for precisely regulation of the measurement condition. Our tribometer is also rotary for a simpler mechanical design whereas the previous studies used linear scanning movements.



(a) Top view



(b) Bottom view

Figure 3.1: Rotary tribometer.

Our tribometer consists of a DC motor (RB-35GM, DnJ, Korea) with a touch pen attached to its shaft using a holder and a six-axis force/torque sensor (Nano 17, ATI Technologies, USA) placed under the tablet (Fig. 3.1). The length of the pen holder is 6 cm, which makes the rotation radius 3 cm. A balancing beam with a counterweight on the opposite end of the touch pen is used to adjust the normal pressure. The beam is pivoted at the middle to provide free vertical movement. The counterweight mass ($\simeq 470\,\mathrm{g}$) is selected to approximate the normal pressure of human hand. The rotation velocity ($\simeq 6.5\,\mathrm{cm/s}$) is chosen for the usual human hand velocity during surface scanning. The force data is sampled at 10 KHz using a 16-bit data acquisition board (NI USB-6251, National Instruments, USA).

3.1.2 Output Characteristics

To characterize the output friction force of Feelscreen, we customized an Android application, originally developed by Senseg, that aligned a number of vertical edges in the landscape orientation (Fig. 3.2). When each edge was crossed by the rotating touch pen of the tribometer, a haptic grain was played back with the designated intensity. A representative data of the measured tangential and normal forces is presented in Fig. 3.3 (haptic grain EDGE-SOFT; intensity 1.0). The upper panel shows that whenever an edge was crossed by the touch pen, a vibratory tangential force occurred with the peak-to-peak (p-p) amplitude of approximately 0.25 N (averaged over 50 largest p-p amplitudes). Vibratory forces were also observed in the normal direction, but their p-p amplitude was much lower (less than 0.05 N). Therefore, the tangential force should be the dominant sensory cue for the perception of haptic effects. These output behaviors are in

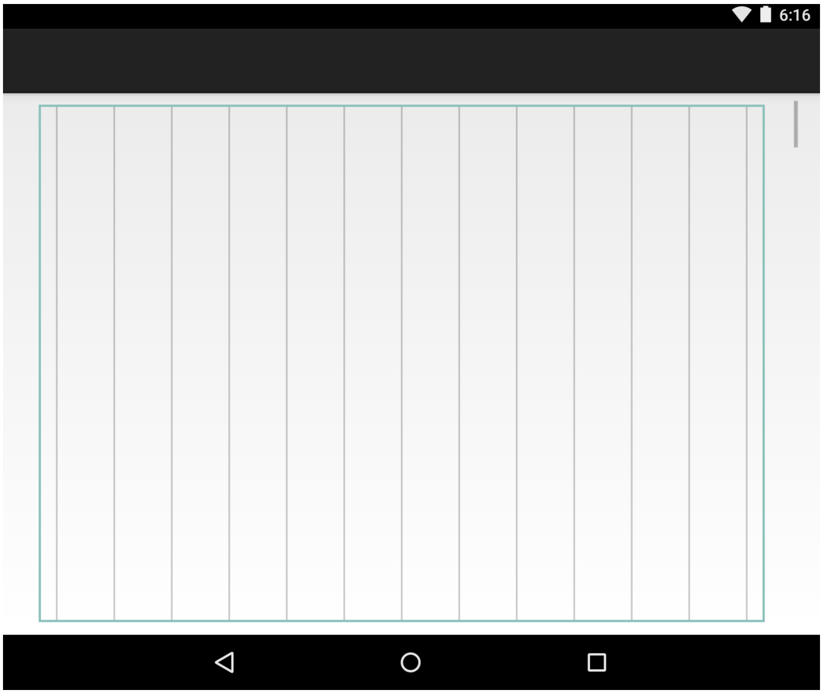


Figure 3.2: Graphical user interface for device characterization. When each vertical line is crossed, a haptic grain is rendered.

good agreement with those reported in the previous related work [13].

We then identified the relationship between input intensity and the magnitude of output tangential force. While changing the input intensity from 0.1 to 1.0 (step size 0.1), we collected 20 s of force data (corresponding to seven full rotations of the tribometer's shaft). After applying a low-pass filter with a cutoff frequency of 500 Hz, we computed the p-p amplitudes of the tangential force and then averaged the 50 largest. The mean p-p amplitude showed a quadratic relationship to the input intensity, as shown in Fig. 3.3(c) (haptic grain EDGE-SOFT). Assuming that input voltage to the electrostatic film of Feelscreen is

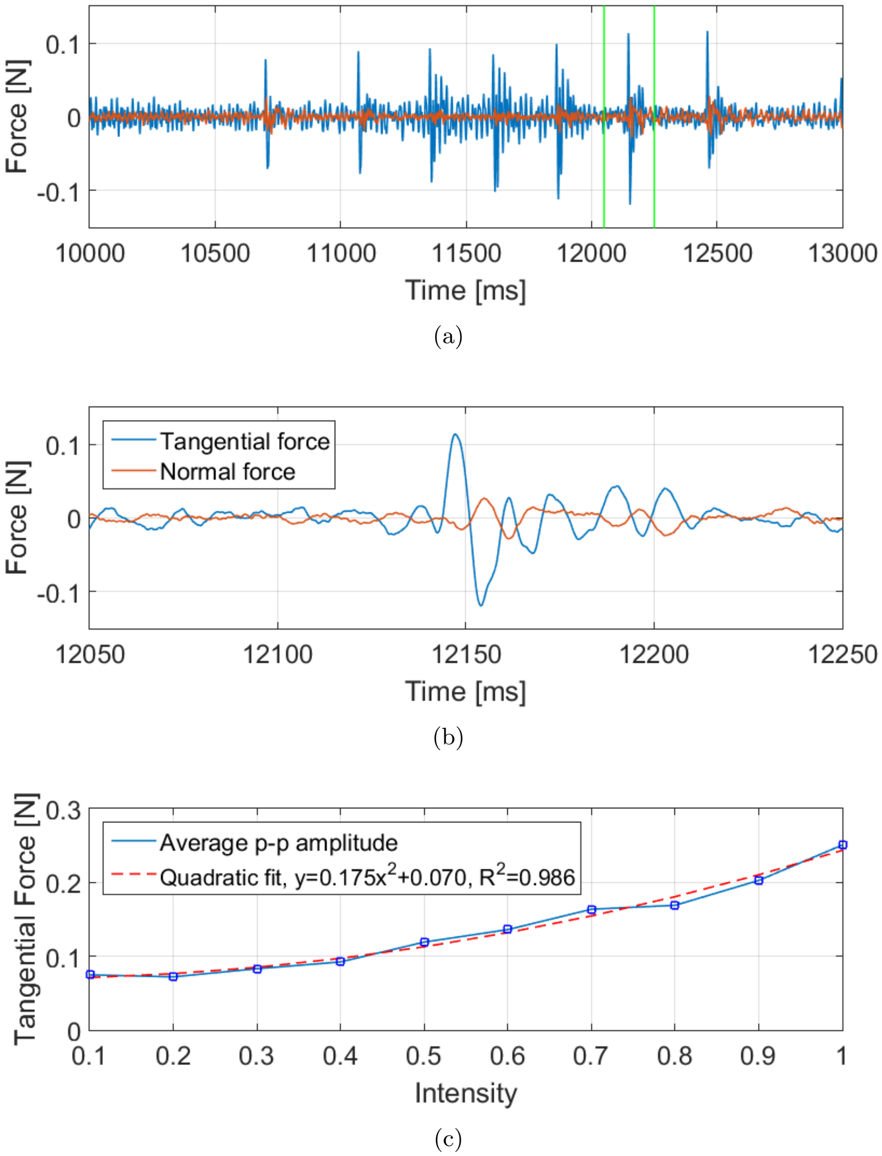


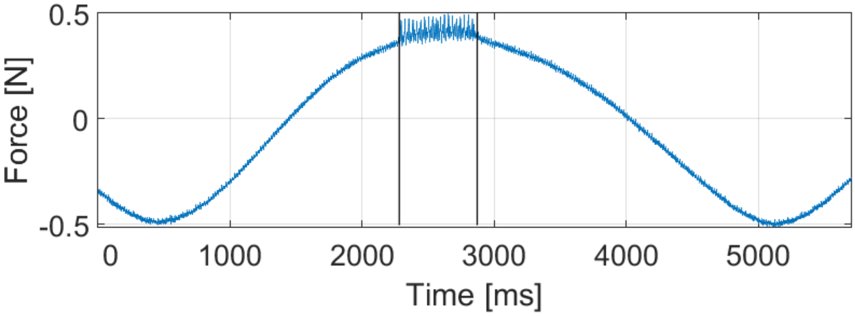
Figure 3.3: Example of force measurements. Blue: tangential force and red: normal force. (a) Raw data. (b) Detail from the region highlighted with two vertical lines in (a). (c) Peak-to-peak amplitude of tangential force vs. input intensity.

linearly proportional to input intensity, this result conforms to the classic theory of electrovibration that the output force magnitude is in proportion to the square of input voltage [5].

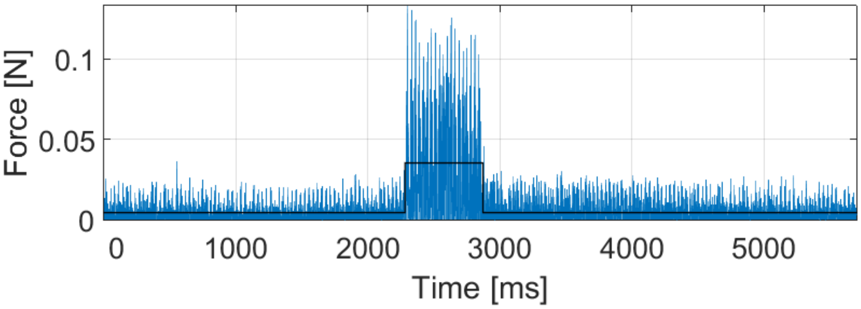
We also looked at the static behavior of friction force. For measurements, the touchpen was rotated on the Feelscreen surface by our tribometer. The surface was divided into eight equal segments, and the EDGE-SOFT haptic grain was enabled for only the first segment (between 0° and 45°) and disabled for the other segments (Fig. 3.4(a)). We computed two average forces from the segments when electrovibration was on and off (Fig. 3.4(b)) and used their difference to represent the increase of static friction force. This procedure was repeated for each input intensity between 0.1 and 1.0 (step size). The average increases of static friction force are shown in Fig. 3.4(c) for input intensities. The relationship was quadratic again, as was for the vibratory friction force.

3.2 Formative user study

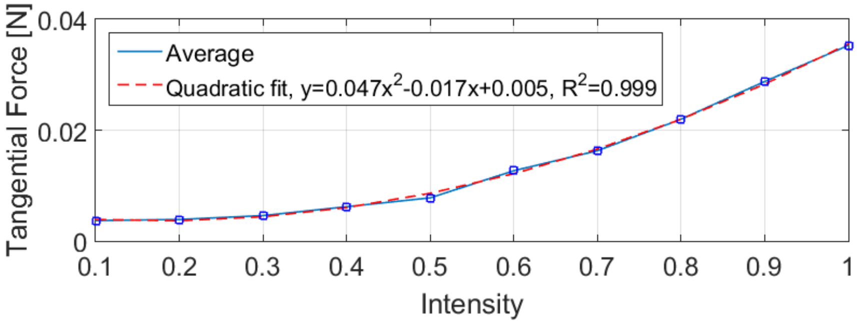
As stated earlier, this perceptual experiment aimed to assess how well users can recognize primitive 3D geometrical shapes when they are provided with depictions of the shapes using only the friction force produced by an electrovibration display. This idea was motivated by the prior work [2] that had demonstrated that rendering tangential force alone can be extremely effective in recognizing geometrical shapes such as bumps and holes. This method was also implemented using a force-feedback haptic interface for inclusion in the experiment as the baseline.



(a) Raw force data



(b) Filtered (blue) and averaged (black)



(c) Quadratic fit

Figure 3.4: Static behavior of the electrostatic friction display. (a) Raw tangential force data. (b) After filtering, the data were averaged to estimate the static friction force increase. (c) Static friction force increase vs. input intensity.

3.2.1 Force Profiles

Following [2], we designed two basic 3D geometries, Gaussian bump and hole, as well as a flat surface for the experiment. The Gaussian profiles had a length of 5 cm and a height of 0.8 cm. They were computed using (3.1) with $\mu = 0$ and $\sigma = 0.5$:

$$y(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{3.1}$$

An exemplar Gaussian bump is shown in Fig. 3.5. The bumps and holes used in the experiment had a width of approximately 2.6 cm.

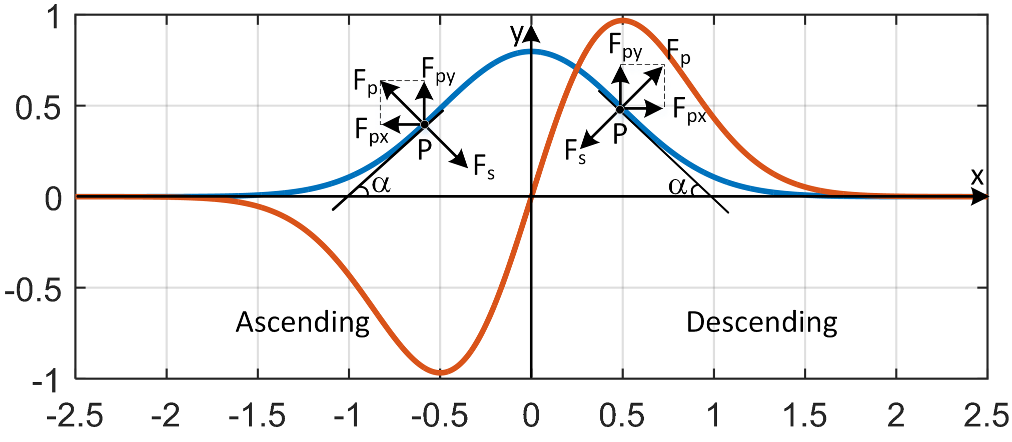


Figure 3.5: Gaussian bump (blue) and the corresponding force profile (red) taken from [2]. The scanning direction is from left to right.

For Force-Feedback Device

To compute force profiles for Gaussian bumps and holes, we followed the footstep introduced in [2]. Assuming the user applies force $\mathbf{F_s}$ at position $\mathbf{p}(x, y)$

when exploring a friction-less physical surface, the surface returns $\mathbf{F_p} = -\mathbf{F_s}$ (Fig. 3.5). From the surface slope at the contact point, the relation between tangential (F_{px}) and normal (F_{py}) components of the returned force can be expressed by

$$\begin{aligned} F_{px} &= -F_{py}\tan(\alpha(x)), \\ \tan(\alpha(x)) &= \frac{dy}{dx} = -\frac{1}{\sigma^2}xy, \end{aligned} \tag{3.2}$$

where $\alpha(x)$ is the angle of $\mathbf{p}(x, y)$.

The normal force F_{py} applied by the user was measured using a force sensor in [2]. However, impedance-type force feedback devices and electrostatic displays generally do not have a force or a pressure sensor. Hence, we assumed in our experiment that $F_{py} = 1\,\mathrm{N}$ based on pilot experiments we conducted using the human hand. An example of the computed tangential force profile for a Gaussian bump is plotted in Fig. 3.5. When the scanning direction is from left to right (in the direction of positive x-axis), the tangential force F_{px} resists the movement during ascending and assists the movement during descending. The force changes its direction at zero slope, right at the summit of the bump.

A computer program was developed using CHAI3D to render the computed force profiles with a force-feedback device using two types of algorithms based on force field and friction, respectively. In the force field-based algorithm, the tangential force profile is directly sent to the force-feedback device. In the friction-based algorithm, the dynamic friction coefficient of a virtual surface is adjusted based on the force profile using a mapping explained in Section 3.2.1.

Five experimental conditions were formed by combining three geometries (bump, hole, and flat surface) and the two rendering algorithms (force field-

Table 3.1: Experimental conditions for force-feedback device.

Condition	Code name
1	FF-BUMP-FR
2	FF-BUMP-FF
3	FF-HOLE-FR
4	FF-HOLE-FF
5	FF-FLAT

FF: force-feedback device, FR: friction-based algorithm, and FF: force field-based algorithm.

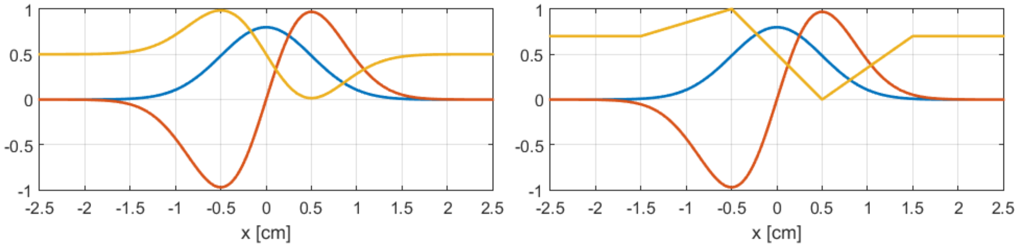
Table 3.2: Experimental conditions for electrostatic device.

Condition	Code name
1	EV-BUMP-IP0.5-HG1
2	EV-BUMP-IP0.7-HG1
3	EV-BUMP-IP0.5-HG2
4	EV-BUMP-IP0.7-HG2
5	EV-HOLE-IP0.5-HG1
6	EV-HOLE-IP0.7-HG1
7	EV-HOLE-IP0.5-HG2
8	EV-HOLE-IP0.7-HG2
9	EV-FLAT-HG1
10	EV-FLAT-HG2

EV: electrovibration, IP: intensity profile, and HG: haptic grain.

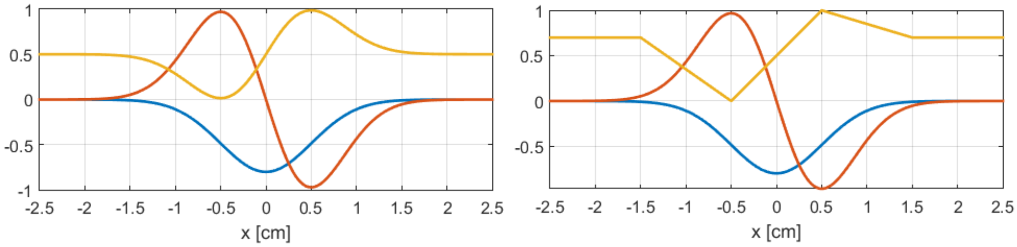
based and friction-based), as summarized in Table 3.1. Only the friction-based algorithm was used for the flat surface.

For Electrostatic Display



(a) Bump, IP0.5.

(b) Bump, IP0.7.



(c) Hole, IP0.5.

(d) Hole, IP0.7.

Figure 3.6: Profiles for the electrostatic display. Blue: geometry [cm], red: force [N], and orange: intensity.

The tangential force profiles for a force-feedback device cannot be rendered using an electrostatic display since it cannot provide the active force assisting movement, e.g., when $F_{px} > 0$ in Fig. 3.5. This is the fundamental limitation of friction displays that are inherently passive. To handle this problem, we linearly map the normalized force of a force profile from -1.0 N to 1.0 N to the input intensity of the Feelscreen tablet from 1.0 to 0.0. This maps -1.0 N to the maximum friction, 0 N to the friction of the half intensity, and 1.0 N to the minimum friction (that of the touchscreen). This is the same technique used in [14]. Since the input

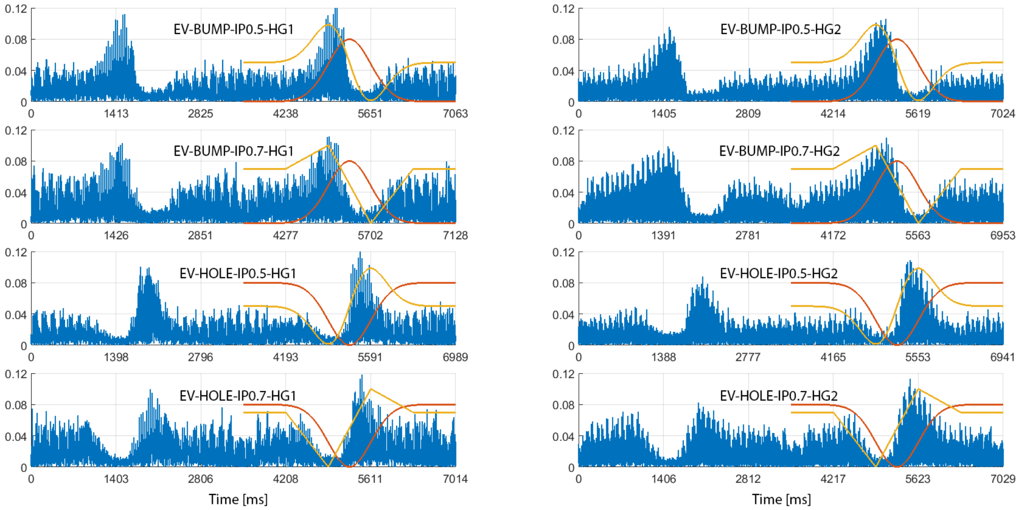


Figure 3.7: Measured friction force profiles for each experimental condition. Blue: measured force profile [N] (only absolute values are shown for clarity), red: geometry profile, orange: intensity profile (scaled to show the trend).

intensity profile has the offset of 0.5, we call this method IP0.5 (see Fig. 3.6a and c). A similar mapping was also used to implement the friction-based rendering for a force-feedback device.

According to the results of device characterization (Fig. 3.3(c)), the actual electrostatic friction for the input intensity of 0.5 is lower than 50% of the full scale force. The minimum friction force is about 0.075 N and the maximum is 0.25 N, and the half full scale force, 0.16 N, occurs around the input intensity of 0.7. This observation led to the design of another intensity profile IP0.7 that uses 0.7 as the offset. For the mapping, we approximate the Gaussian force profiles with piecewise linear intensity profiles (Fig. 3.6b and d).

We developed an Android application using min3d (an open-source engine based on OpenGL ES) to graphically render geometrical profiles (although hidden from the participants) and also to render intensity profiles in response to the user's touch position. Ten experimental conditions were prepared by combining three geometries (bump, hole, and flat surface) with the two intensity profiles (IP0.5 and IP0.7) and two haptic grains (HG1: EDGE-SOFT and HG2: AREA-GRAIN). The two haptic grains were chosen based on pilot experiments. A constant force profile with the maximum intensity 1.0 is used for the flat surface. The ten experimental conditions are summarized in Table 3.2.

The electrovibration stimuli measured using the tribometer are shown in Fig. 3.7 for the eight experimental conditions for bumps and holes. The geometry and intensity profiles are also shown for reference. It is clear that the induced electrostatic friction forces were in good match with the corresponding intensity profiles. The friction force patterns are clearly distinguishable between bumps and holes. The friction forces rendered using IP0.7 resulted in more symmetric profiles than those rendered using IP0.5.

3.2.2 Participants

Twelve participants (9 male, 3 female; M 22.7 years, SD 2.6 years) were recruited using an on-line public announcement. All of them were students enrolled at the authors' university. None of them reported noteworthy previous experiences of using kinesthetic haptic interfaces or electrostatic displays. They signed on an informed consent prior to the experiment. Each participant was compensated 10,000 KRW (9 USD) for their help.

3.2.3 Procedure

In the experiment, we used a PHANToM (1.0A; Geomagic, USA) as a force-feedback device and the Feelscreen tablet as an electrostatic display. Participants were randomly divided into two groups. The participants in group G1 first performed the five experimental conditions in Table 3.1 with the PHANToM, and then the ten experimental conditions in Table 3.2 with the Feelscreen tablet. These were switched for the participants in group G2. The order of the experimental conditions for the PHANToM and that for the Feelscreen tablet were randomized for each participant.

For each device, the experiment consisted of two parts. Part 1 was for open descriptions; participants were asked to freely describe their percept and experience in writing after exploring each stimulus. Nothing was provided to participants that could bias their perception. Part 2 was for a closed question; for each stimulus, participants chose one of the following four answers: 1) bump, 2) hole, 3) flat surface, and 4) none of them. They were instructed to select the shape that best describes their percept. Part 1 was performed first, followed by Part 2 using the same device after a short break.

During the experiment, the haptic device was placed inside a box with frontal access to a participant. A curtain covered the box to block the participant's view to prevent them from obtaining any visual cues. The experimenter could see the device from the back of the box and provided occasional guidance to the participant's pose and scanning speed when necessary. For the experiment with the Feelscreen tablet, participants were asked to hold a touch pen vertically and scan the surface from left to right. Each of the ten experimental conditions was presented only once. For the experiment with the PHANToM, the same touch pen was attached to the last vertical link of the PHANToM, and a seven-inch tablet was placed under the touch pen to enable similar scanning experiences. Participants were asked to hold the touch pen vertically and scan the surface from left to right. Each of the five experimental conditions was repeated twice. The numbers of bumps, holes, and flat surfaces were not known to participants in order to prevent guessing based on counting.

Prior to the experiment, participants were given enough time to practice and become familiar with the system. During the experiment, they were allowed to take rest whenever necessary. Participants' scanning velocity and vertical pressure were not controlled. They were free to adjust their own velocity and pressure for better perception; however it was supervised by the experimenter. The experiment took approximately one hour to finish for each participant.

3.3 Results and Discussion

3.3.1 Open Descriptions

We compiled the participants' answers collected in the first part of the experiment. No noticeable differences were observed between the participants of group G1 and G2 in the open descriptions.

Most of the participants described the sensations of the force feedback rendered by the PHANToM using geometrical terms and figures. Frequently used words included bump, hole, protrusion, groove, convex or concave shape, ascent, and descent. Only one participant did not use any geometry-related term and instead used material-related terms, e.g., spring. These results reinforce the previous finding of [2] that the lateral force alone can render clearly identifiable primitive 3D shapes.

For the surfaces rendered using electrovibration, the majority of the participants described their sensations with terms related to vibration and friction, and sometimes texture. Only one participant mentioned geometrical terms (hole). It appears that electrovibration alone is not able to elicit strong illusions for the perception of 3D geometric shapes.

The time the participants spent to complete each experimental condition was shorter with the PHANToM than with the Feelscreen tablet. After just three or four scans with the PHANToM, the participants began to write down their descriptions. The Feelscreen tablet usually required six or seven scans for that.

3.3.2 Closed Selections

From the participants' responses collected in Part 2 of the experiment, the average correct recognition ratios of the geometrical shapes were computed for the two devices and are shown in Fig. 3.8. The correct recognition ratio for each device was calculated by dividing the total number of correct answers from all participants received in each experimental condition by the total number of answers in that condition. As expected, higher recognition performance was achieved with the PHANToM than with the Feelscreen tablet—91% vs. 64%. The Kruskal-Wallis test showed that the difference between the two devices was statistically significant ($p < 0.001$). The same test was performed between the two participant groups G1 and G2, but their recognition performance difference was not statistically significant ($p = 0.88$). These results suggest that when participants were given explicit guidance, they were able to associate the electrovibration patterns to the primitive 3D shapes at well above the chance level (25%). However, there existed a substantial performance difference (27%) from the best performance enabled by active force feedback.

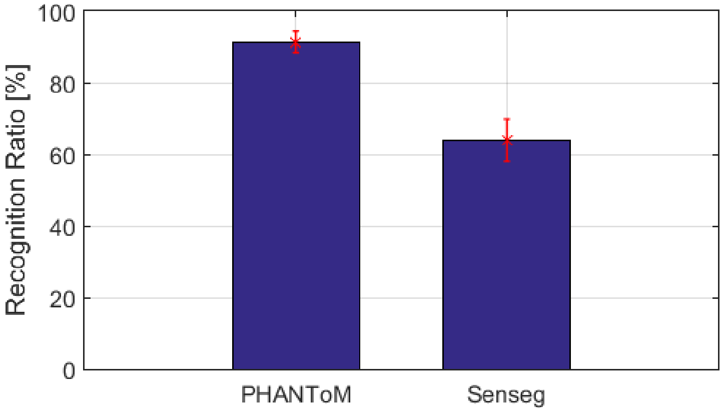


Figure 3.8: Correct recognition ratios using the PHANToM (mean 91%) and the Feelscreen tablet (mean 64%). Error bars show standard errors.

Fig. 3.9 shows the average correct recognition ratios measured with the PHANToM for each experimental condition.[1] On average, bumps resulted in a higher ratio than holes (94% vs. 89%), which can be seen by comparing the ratios of conditions 1 and 2 and those of conditions 3 and 4. In addition, the

---

[1] Only one ratio was computable for each experimental condition. Hence no error bars are shown in Fig. 3.9 and 3.10. No statistical tests were performed with the data for the same reason.

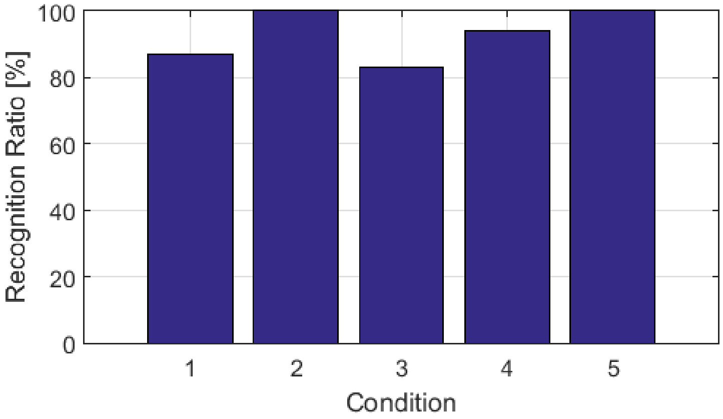


Figure 3.9: Correct recognition ratios with the PHANToM for each experimental condition (see Table 3.1).

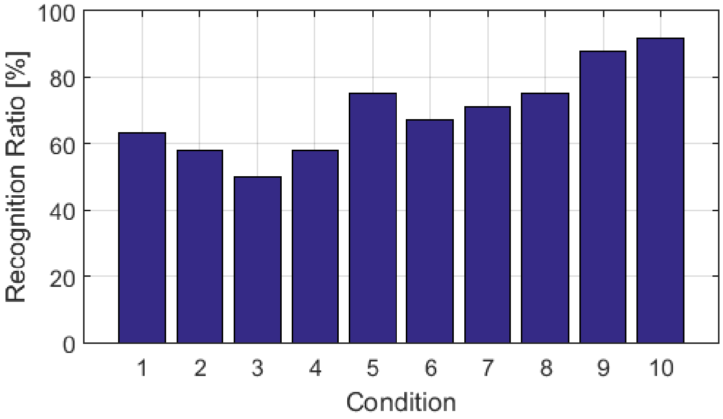


Figure 3.10: Correct recognition ratios with the Feelscreen tablet for each experimental condition (see Table 3.2).

force field-based algorithm showed higher performance than the friction-based algorithm (97% vs. 85%; compare the ratios of condition 1 and 3 to those of condition 2 and 4).

Similarly, the average correct recognition ratios measured with the Feelscreen tablet are presented in Fig. 3.10 for each experimental condition. The most prominent observation is that holes (conditions 5–8) gained more correct recognition than bumps (conditions 1–4) with 72% vs. 57%. The performance difference caused by the two different intensity profiles was negligible (IP0.5 66% vs. IP0.7 65%), and so was the recognition accuracy difference between the two haptic grains (HG1 69% vs. HG2 70%).

3.3.3 Summary and Discussion

The experimental results allow us to draw the following conclusions to the two research questions of this study:

Q1 *Can users identify primitive 3D features, such as bumps and holes, from electrovibration alone without any visualization?*

The answer is negative if no guidance or context implying association to geometric shapes is provided.

Q2 *How close is the recognition performance to that of the case using an active kinesthetic interface?*

If a hint to geometric shapes is given, users can associate electrovibration patterns to geometrical shapes at well above the chance level (64%), but the performance is clearly below the best performance (91%) achievable by active force feedback.

Lateral force feedback using electrovibration has two important differences from active force feedback. First, it conveys clear sensations of vibration, as predominantly mentioned in the participants' open descriptions. This seems to be one of the major reasons that preclude users from associating electrovibration patterns to geometrical shapes unless guided explicitly. Second, electrovibration does not allow the rendering of active tangential force that assists the movement when the gradient of a surface profile is negative. Although we tried to imitate this behavior using only friction, it seems that the effectiveness of that approach has a room for further improvement.

The similar performance between the two haptic grains HG1 and HG2 is an indication that the type of haptic grain is not a main factor for geometry recognition, although they may provide different feelings. Delivering noticeable friction fluctuations according to the intensity profiles appears to be sufficient. The same can be said to the effect of intensity profile on the basis of the similar recognition ratios of the two intensity profiles IP0.5 and IP0.7.

3.4 Conclusions

In this study, we compared an electrostatic tablet and a kinesthetic haptic interface in terms of their performance for rendering 3D shapes using only tangential force. Since the commercial electrostatic device used was a black box to us, we first characterized its input-output behavior of generating friction force. Then we carried out a perceptual experiment that assessed the user's recognition performance of primitive 3D shapes based on the tangential stimuli presented by the electrostatic tablet and a force-feedback interface. Experimental results demonstrated that users are not able to absolutely associate electrovibration patterns to the geometrical shapes without any contextual information. However, when such guidance was given, participants showed moderately high recognition performance of the primitive shapes, which is promising for the possibility of improving the user experiences of 3D visual content with the provision of electrovibration. The results obtained with the force-feedback device were used as the reference for the best performance.

Chapter IV.
Generalized Gradient Algorithm For Rendering 3D Meshes

The formative user study in the previous chapter indicated that electrostatic friction displays have the potential to render 3D features by modulating friction force according to the gradient of the profile if sufficient context is provided. However, the algorithm used was for the explicit representation of the profile. This motivated us to generalize the basic algorithm implemented for the formative user study to cover general 3D objects modeled using meshes, as detailed in this chapter.

4.1 Fundamental Algorithm

Suppose that a user explores a 3D object displayed on a touchscreen by scanning it with a finger. We call the finger's position a touch point (x, y) defined in the 2D touchscreen coordinates. For lateral force feedback, this 2D touch point is mapped to the 3D touch point (x, y, ∞) in the 3D world coordinate frame of the 3D object; see points marked by X in Fig. 4.1. From each 3D touch point, a vertical ray is cast along the z-direction toward the 3D object for collision detection. If any collisions are detected, we take the collision point with the greatest height (z-coordinate) and call it a hit point for the ray; see points marked by O in Fig. 4.1.

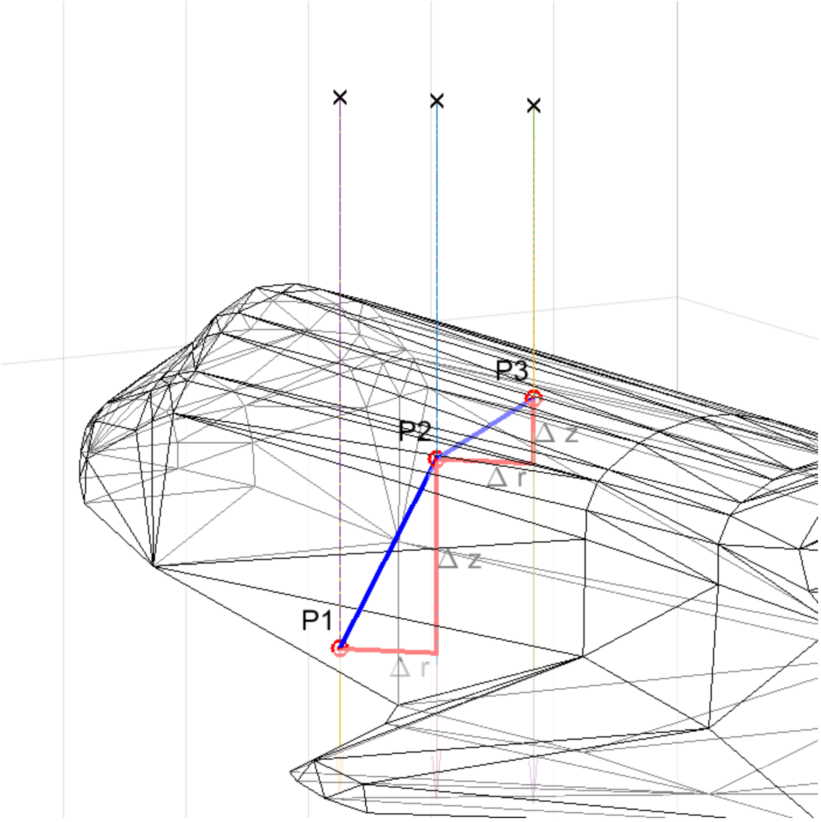


Figure 4.1: Variables for the generalized lateral force rendering algorithm.

We estimate the surface gradient that the user's finger experiences by computing the linear slope between the two consecutive hit points of the current and previous frames. Let the current hit point be $P2 = (x_2, y_2, z_2)$ and the previous hit point be $P1 = (x_1, y_1, z_1)$ in Fig. 4.1. Then the surface gradient at $P2$ is approximated by

$$\tan(\alpha) = \frac{\Delta z}{\Delta r}, \tag{4.1}$$

$$\Delta z = z_2 - z_1, \tag{4.2}$$

$$\Delta r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{4.3}$$

Recall that the lateral force is obtained from

$$F_{pxy} = -F_{pz} \tan(\alpha). \tag{4.4}$$

With $F_{pz} = 1\,\mathrm{N}$ as in the formative user study,

$$F_{pxy} = -\frac{\Delta z}{\Delta r}. \tag{4.5}$$

As we have shown earlier, the relation between input intensity and output force is quadratic. Also to preserve force direction, we consider a signed square root of lateral force as follows:

$$F^{sq} = \mathrm{sgn}\,(F_{pxy}) \sqrt{|F_{pxy}|}, \tag{4.6}$$

and then map F^{sq} to the input intensity I.

An important issue here is that F^{sq} can have an arbitrary value depending on the object geometry and the scanning direction and velocity of the user's finger, but the tangential force that an electrostatic display can generate is limited. Hence, we need to assume that F^{sq} is also bounded, i.e., $F^{sq}_{min} \leq F^{sq} \leq F^{sq}_{max}$, and then linearly map this interval into to the input intensity $I \in [1.0, 0.0]$ of the haptic grain to generate. Note that F^{sq}_{min} (resisting force; note its sign) must provide the maximum friction ($I = 1.0$) by the electrostatic display whereas F^{sq}_{max} (assisting force) should exert the minimum friction ($I = 0.0$). This can be elaborated by considering the empirical observation that I lower than 0.3 does not produce perceptually clear lateral force. That is, we linearly map F^{sq} in $[F^{sq}_{min}, F^{sq}_{max}]$ to I in [1.0, 0.3]:

$$I = \frac{0.7F^{sq} + 0.3F^{sq}_{min} - F^{sq}_{max}}{F^{sq}_{min} - F^{sq}_{max}}. \tag{4.7}$$

If $F^{sq} \leq F^{sq}_{min}$, $I = 1.0$, and if $F^{sq} \geq F^{sq}_{max}$, $I = 0.3$.

4.1.1 Computing Minimum and Maximum Force

The generalized lateral force rendering algorithm is straightforward, but how to determine $(F^{sq}_{min}, F^{sq}_{max})$ requires some careful considerations. In the formative user study, $F^{sq}_{min} = -1$ and $F^{sq}_{max} = 1$ since the minimum and maximum of lateral force F_{px} were -1 N and 1 N, respectively, for the 2D Gaussian profiles. In general 3D cases, minimum and maximum lateral forces cannot be obtained easily. They depend on the scanning direction and speed of the user's finger and also the geometry of the 3D object. If $(F^{sq}_{min}, F^{sq}_{max})$ is not set properly, the rendering will use either a narrow range of I or the minimum or maximum friction excessively, being unable to express the geometric changes of the 3D object appropriately.

To determine an effective range for $(F^{sq}_{min}, F^{sq}_{max})$, we simulate the haptic rendering process by scanning the object with various velocities and directions. This process results in a large number of F^{sq} values. Then we use $(F^{sq}_{min}, F^{sq}_{max}) =$

$(\mu - 2\sigma, \mu + 2\sigma)$, where μ is the mean and σ is the standard deviation. This corresponds to the 95% confidence interval if F^{sq} follows a Gaussian distribution.

We empirically confirmed that this simple method generally works well for many 3D models. It is also demonstrated in the summative user study in Section 4.2. There can also be good alternatives depending on the distribution of F^{sq}.

4.1.2 Edge Emphasis

The generalized lateral force rendering algorithm provides haptic feedback regarding the convexity or concavity of a 3D object. However, sharp edges on the surface may be rendered inadequately if their neighbor faces result in F^{sq} close to F^{sq}_{min} or F^{sq}_{max}, or outside $(F^{sq}_{min}, F^{sq}_{max})$. In such cases, the device output is saturated, and the lateral force rendering algorithm alone cannot render sharp tactile sensations for the edges. To remedy this problem, we detect the edges of a 3D object and emphasize them as follows.

Many edge detection algorithms are available in computer vision and graphics [61, 62], and those taking a mesh as input are pertinent to our purpose [63]. Most of such algorithms are based on the notion of *dihedral angle*: the angle between two planes that share an edge. Dihedral angle roughly approximates the principle curvature of the surface at the edge [64]. An edge with a dihedral angle greater than a given threshold is considered as a sharp edge.

In parallel with the gradient-based lateral force calculation, we also compute at each frame the dihedral angle θ from the polygon to which the previous hit point belongs to the polygon on which the current hit point lies. θ can be easily calculated from the normal vectors of the two polygons. An edge is convex if

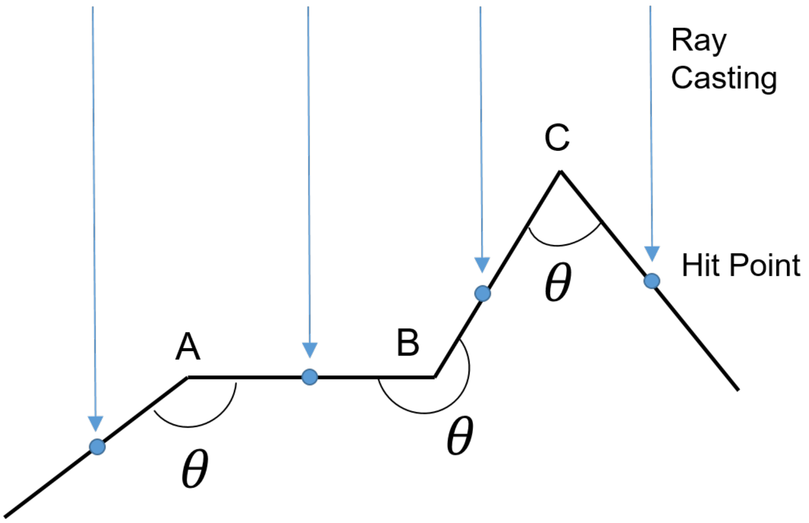


Figure 4.2: Dihedral angles. Thick lines represent the cross-section of polygons.

$\theta < 180°$ (A and C in Fig. 4.2) and concave if $\theta > 180°$ (B in Fig. 4.2). We consider only convex edges since concave ones are not reachable in reality. Further, a convex edge becomes sharper if the dihedral angle θ is decreased from 180° to 0° (compare A and C in Fig. 4.2). Hence, we represent the sharpness S of a convex edge by

$$S = -\frac{\theta - 180°}{180° - \theta_{min}}, \tag{4.8}$$

where θ_{min} is the smallest dihedral angle of the object. This function maps $\theta = 180°$ to $S = 0$ and $\theta = \theta_{min}$ to $S = 1$.

To emphasize convex edges, we play a haptic grain of different type (EDGE-TICK) with intensity I_e. This is in comparison to the AREA type haptic grain (AREA-EVEN) used for surface profile rendering. The two haptic effects generate easily distinguishable frictional stimuli (Fig. 4.3). I_e is determined by linearly mapping the sharpness S to the intensity interval [0.3, 1.0]:

$$I_e = 0.7S + 0.3. \tag{4.9}$$

An example of edge emphasis is given in Fig. 4.4.

4.2 Summative User Study

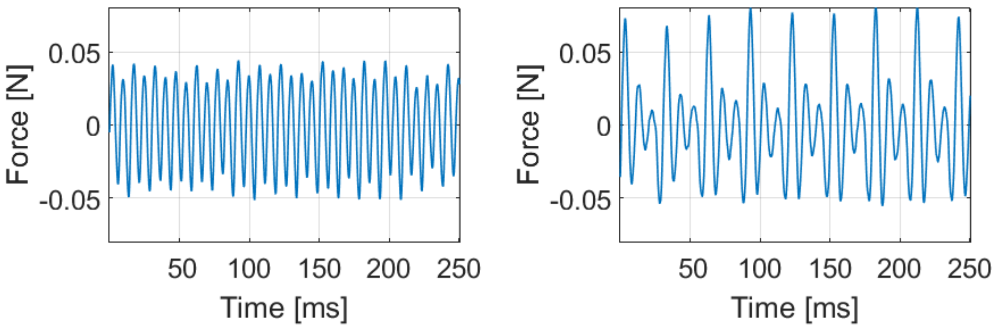
We conducted another user study to evaluate the effectiveness of our generalized gradient-based lateral force rendering algorithm. The task was to recognize 3D objects presented to the user with limited visual information. This task was motivated by the finding of the formative user study that frictional electrovibration can deliver 3D geometric information if its context is explicit to users. Some visual stimuli were designed to be difficult for visual recognition. Our aim was to assess the extent to which our electrovibration rendering algorithm can facilitate the recognition task in such situations. This study was approved by the IRB of POSTECH (PIRB-2016-E015).

4.2.1 Methods

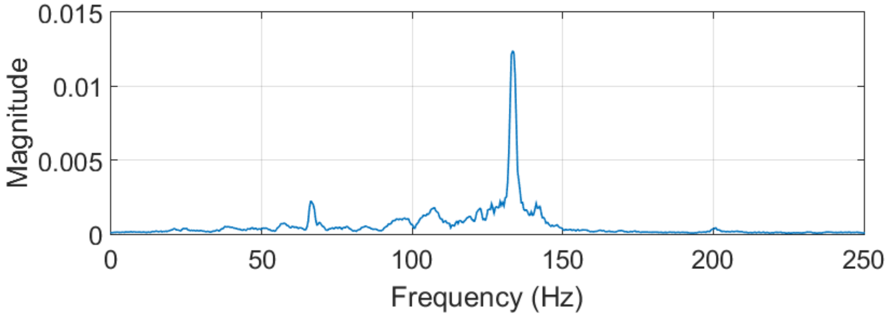
Stimuli

We used two 3D objects, bumps and holes, each in two different profiles, Gaussian and square frustum, in this experiment (Fig. 4.5). Detailed equation and dimension are given in Fig. 4.5 for the bumps. The holes were made by inverting the corresponding bumps.

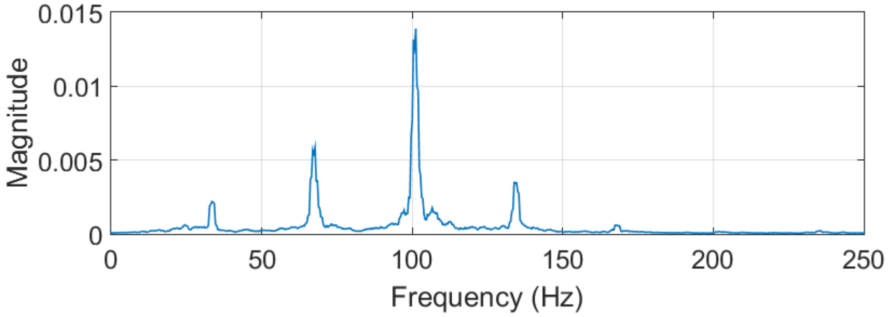
One visual scene consisted of four bumps and holes, but their profiles were the same (all Gaussians or frustums). Varying the numbers of bumps and holes



(a) AREA-EVEN (b) EDGE-TICK

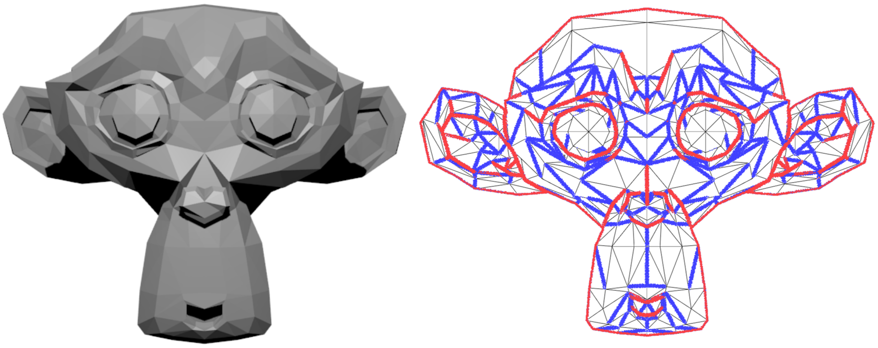


(c) AREA-EVEN



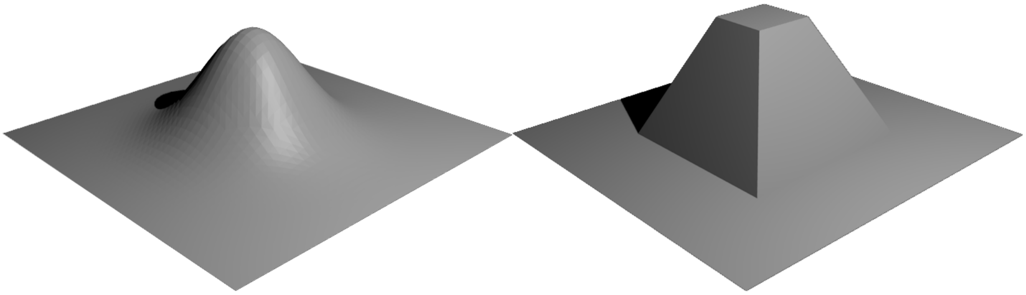
(d) EDGE-TICK

Figure 4.3: Plots of haptic grain AREA-EVEN (a) and EDGE-TICK (b), and their spectrums (c and d). AREA-EVEN has a single main spectral component, and this effect generates smooth electrostatic force. EDGE-TICK includes multiple spectral components, which makes it feel rough and bumpy [3]. EDGE-TICK conveys an impression of sudden impact and is suitable for rendering sharp edges.



(a) (b)

Figure 4.4: (a) 3D mesh of a Monkey model. (b) Results of edge emphasis. All convex edges are highlighted in blue. Among them, those with $\theta < 135°$ (relatively sharp ones) are marked in red.



(a) (b)

Figure 4.5: 3D shapes used in the summative user study. (a) Gaussian bump: width 6, length 6, height 1.8, $z(x,y) = \frac{8}{1.4\pi}\exp(\frac{-x^2-y^2}{1.4})$. (b) Square frustum bump: width 6, length 6, height 2, base square 3×3, top square 1×1. All dimensional units are cm.

resulted in five different configurations (Fig. 4.6): 0B4H (0 bump and 4 holes), 1B3H (1 bump and 3 holes), 2B2H (2 bumps and 2 holes), 3B1H (3 bumps and 1 hole), and 4B0H (4 bumps and 0 hole). Which were bumps or holes out of the four positions was determined randomly for each configuration, but once made the same scene was used for all participants. For example, in configuration 1B3H, a bump can be placed in one of the four locations (top-left, top-right, bottom-left, bottom-right). We chose to fixe the bump to top-right (for Gaussian scenes) and to bottom-right (for square frustum scenes). This was to keep the size of the experiment manageable. We also confirmed that similar visual information was presented regardless of the positions of bumps and holes.

To limit visual information, we controlled several design factors as follows: 1) Orthographic projection was used to render the 3D scenes. A perspective projection may provide sufficient visual information for object identification; 2) Only the top view was shown to the user. Looking at objects from other angle may enable sufficient visual information even with orthographic projection; 3) Two lighting conditions were applied: a vertical spotlight and a directional light rotated (45°, 45°, 0°) around the $x - y - z$ axes. They create very different shadows depending on the object configuration in the scene. 4) All objects were rendered in gray; and 5) No textures were added to the objects. Combining all these factors, 20 ($= 2$ shapes $\times$ 2 lights $\times$ 5 configurations) different scenes were designed (Fig. 4.6).

Participants

Twenty participants (14 male, 6 female; M 22.7 years old, SD 3.0 years) were recruited for this study. They were divided into two groups of 10: G1 (7 male, 3 female, M 22.3, STD 3.0) and G2 (7 male, 3 female, M 23.1, STD 3.1). No participants reported having sensorimotor impairments that could affect experimental results. The experiment took about 30 min. Each participant was paid 5000 KRW ($\simeq$ 4.5 USD).

Procedure

We used a Senseg Feelscreen as the electrostatic tactile display. An Android application (Fig. 4.7) was developed using Unity (version 5.32) [65] for both visual and haptic rendering with an update rate of 60 Hz. The sampling rate for tactile signal reconstruction was much higher and determined internally by the Feelscreen library.

Each participant completed two sessions of trials. In the session for the *Visual* task (V), participants were asked to only look at the objects and provide answers as to whether they were bumps or holes. No tactile feedback was provided. In the session for the *Visual+Haptic* task (V+H), participants were told to also touch the objects and then provide their answers. Tactile feedback was presented using our generalized lateral force rendering algorithm.

In the beginning of each session, participants were given enough time to become familiar with the application. For group G1, a session for V+H was conducted first, followed by a session for V. For group G2, the order was reversed for balanced order. In each session, the 20 scenes were presented to participants once, and their order of presentation was randomized for each participant.

4.2.2 Results

We first computed the correct recognition ratios for each participant in each group. A pairwise t-test found no significant difference between the two groups

Table 4.1: Results of Four-way Within-Subject ANOVA.

Factor	F-value	p-value
Task*	F(1,19)=27.39	<0.0001*
Light*	F(1,19)=4.56	0.0460*
Geometry	F(1,19)=0.04	0.8492
Configuration	F(4,76)=1.76	0.1462
Task×Light*	F(1,19)=7.1	0.0153*
Task×Geometry	F(1,19)=0.87	0.3636
Task×Configuration	F(4,76)=0.36	0.8373
Light×Geometry	F(1,19)=1.15	0.2967
Light×Configuration	F(4,76)=0.91	0.4650
Geometry×Configuration	F(4,76)=0.80	0.5261
Task×Light×Geometry	F(1,19)=2.17	0.1575
Task×Light×Configuration	F(4,76)=2.03	0.0988
Task×Geometry×Configuration	F(4,76)=0.68	0.6084
Light×Geometry×Configuration	F(4,76)=1.96	0.1099
Task×Light×Geometry×Configuration	F(4,76)=0.33	0.8580

G1 and G2 ($t = 0.14, p = 0.7091$). This suggests that the order of the two sessions did not affect the experimental results. Thus we pooled the data of both groups for further analysis.

The experiment had four independent factors: task (V and V+H), geometry (Gaussian and square frustum), lighting condition (spotlight and directional light), and configuration (0B4H, 1B3H, 2B2H, 3B1H, and 4B0H). For statistical analysis, we applied a four-way within-subject ANOVA, and its results are shown in Table 4.1.

Task (under different types of sensory feedback), V vs. V+H, had a statistically significant effect on the correct recognition ratio with a very small p-value. This can be verified by comparing the mean correct recognition ratios between the two tasks shown in Fig. 4.8, with 75% vs. 95% for V and V+H. This result indicates that our shape rendering algorithm using frictional feedback was effective in delivering supplemental shape information when visual rendering alone was not sufficient, improving the correct recognition ratio by 20%.

Between the two lighting conditions, spotlight and directional light, a noticeable difference was also reported ($p = 0.0460$). On average, directional light showed higher recognition performance than spotlight with 88% vs. 81% (Fig. 4.8) by 7% difference. Note, however, the p-value close to the boundary of significance ($\alpha = 0.05$).

Among the many interactions between the main factors, only task×light was statistically significant ($p = 0.0153$). This means that the two light sources had different effects depending on the task, as shown in Fig. 4.9. For task V, light was a crucial factor in which directional light achieved higher performance than spotlight with 83% vs. 67%. However, in task V+H with the added haptic feedback, light was no longer a main player and both light sources showed very similar performance (96% vs. 94%). In addition, although the performance was generally improved from task V to V+H, the increase rate was approximately 2.5 times larger with spotlight than with directional light. This is another support that the frictional electrovibration feedback was quite effective in rendering 3D shape information.

The other main factors and interaction terms did not have significant effects on the recognition accuracy.

Aside from the main factors, we also investigated how bumps and holes were recognized. The performance difference caused by the two different shapes was negligible (hole 86% vs. bump 84%). A pairwise t-test did not find a significant effect of bumps and holes on the recognition ratio ($t = 0.91, p = 0.3402$).

4.2.3 Discussion

Task V+H showed clearly higher performance than task V by the 20% improvement of correct recognition with statistical significance. This is due to the additional haptic information provided in task V+H, but not in task V. Therefore, the summative user study provides unequivocal evidence that our generalized lateral force rendering algorithm is effective in rendering 3D shape using electrostatic frictional force.

We could find an interesting observation on the roles of visual and haptic cues. Shading was constant and determined by the position of the objects in the scene with respect to the light source. For example, for bumps, the back faces (from the light source) were always darker than the front faces. This was not the case for haptic feedback since friction changes depended on the scanning direction. In other words, there was no correlation between shade (visual cue) and friction (haptic cue). Depending on the scanning direction, the dark side of a bump could have either large or small friction. This could have created a conflict between the two sensory cues. In spite of this situation, the recognition ratio in task V+H was very high (95%). This implies that participants relied more on haptic cues than visual cues to resolve the conflict in sensory information.

We also measured the task completion times for each participant in each trial. They were, on average, 213 s and 427 s for task V and V+H, respectively. This was expected since task V+H required manual scanning on the objects.

4.3 Conclusions

In this study, extending the basic algorithm introduced in previous chapter, we have presented a generalized algorithm to estimate surface gradient from the given 3D mesh. The generalized algorithm computes the frictional rendering force by estimating the gradient at the touch point and also emphasizes sharp edges on the surface by rendering perceptually salient friction effects. The generalized algorithm is also computationally inexpensive. We conducted a summative user study to evaluate the effectiveness of our shape rendering algorithm in reducing the visual uncertainty in 3D shape perception. We found that when frictional feedback was provided, the correct recognition performance was notably increased in comparison to when only visual rendering was presented. To our knowledge, our algorithm is among the first that attempted to improve the perception of 3D features displayed on a touchscreen using electrovibration feedback.

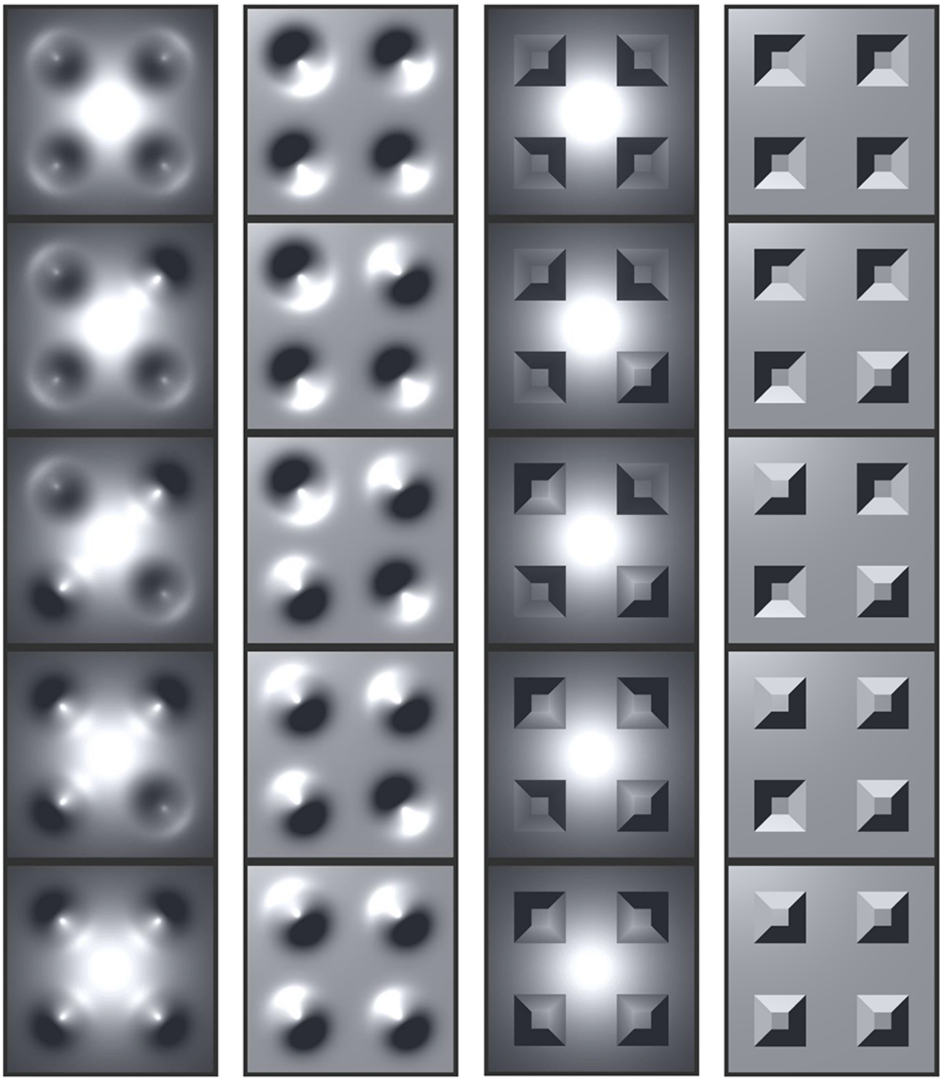


Figure 4.6: Experimental conditions. Twenty scenes were composed by combining two shapes (Gaussian and square frustum), two lighting conditions (spotlight and directional), and five configurations (0B4H, 1B3H, 2B2H, 3B1H, and 4B0H). From left to right: Gaussian-spotlight, Gaussian-directional light, square frustum-spotlight, and square frustum-directional light. From top to bottom: 0B4H, 1B3H, 2B2H, 3B1H, and 4B0H. Some scenes can be easily confused because the direction of directional light in not known to users. For instance, the (2, 2) scene (Gaussian-spotlight, 1B3H) can be mistaken with the (4, 2) scene (Gaussian-spotlight, 3B1H).

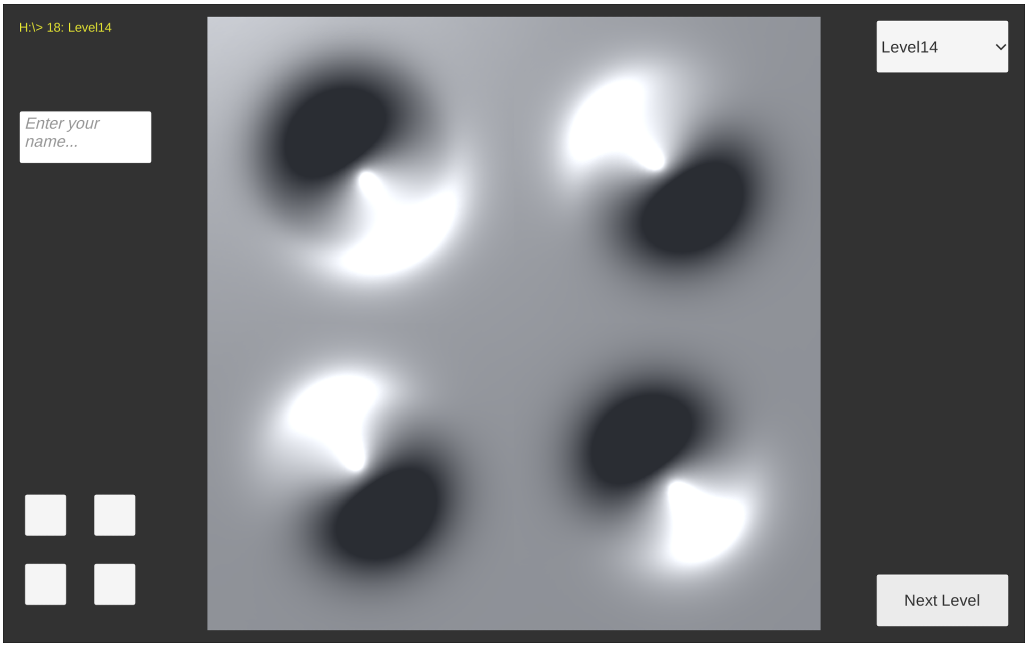


Figure 4.7: Snapshot of the GUI program. In the lower right corner, a button was provided to move to the next scene after providing answers using the four toggle buttons in the lower left corner. Participants checked each toggle button for bump and unchecked it for hole, for the corresponding object in the scene.

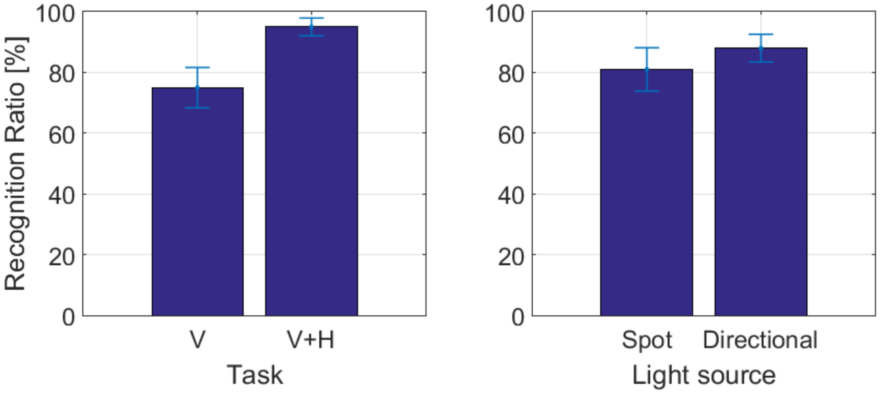


Figure 4.8: Correct recognition ratios for task (V 75% and V+H 95%) and light (spotlight 81% and directional 88%). Error bars show 95% confidence intervals.

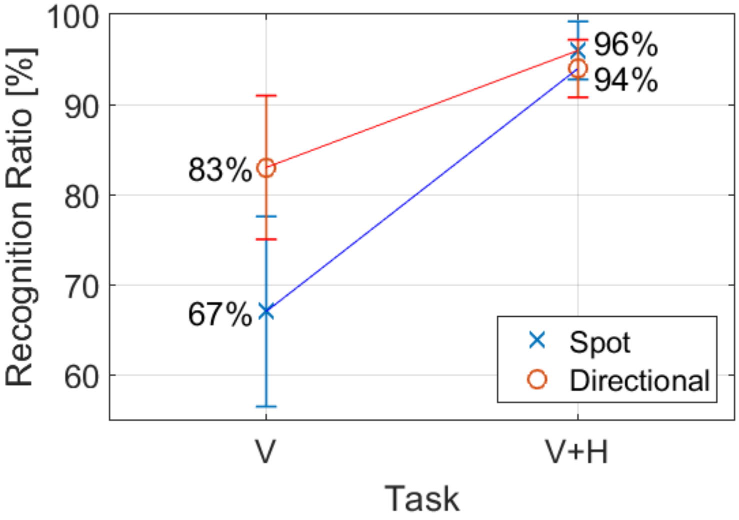


Figure 4.9: Interaction effects between task and light. Error bars show 95% confidence intervals.

Chapter V.
Data-driven Texture Rendering

So far, we have investigated methods to render surface geometry on an electrovibration display. A natural and intuitive extension is to continue with the surface fine texture. Texture rendering has been one of the challenging topics in the haptics community. Realistic haptic texture rendering not only provides richer information but also improves user experience. In this work, we propose using neural networks for data-driven texture rendering on an electrovibration display. While most research groups have considered contact acceleration data as the salient feature of surface texture, in this attempt we instead focus on lateral frictional forces. The main reason stems from the fact that an electrovibration display modulates friction on the sliding medium; thus choosing lateral force to study the textural behavior of surface is more natural and appropriate. We designed a linear motorized tribometer for accurate and easy force measurements from the surface of real texture sample and also that of an electrovibration display.

By actuating the display with a full-band pseudo-random binary signal (PRBS) and collecting the resultant lateral forces, we inversely train a neural network to learn an appropriate input signal from the measured forces. Next we collect forces from real samples and use the trained networks to generate corresponding PRBS-like actuation signals. The generated signals are applied to the display and once more the frictional forces are collected. A comparison between the real and virtual forces in frequency domain shows promising results and reveals the capabilities and limitations of the proposed technique.

5.1 Data Collection

This section describes the hardware we use to collect lateral frictional forces. We then introduce the use of a capacitive touchscreen panel as an electrovibration display.

5.1.1 Tribometer

We developed a motorized linear tribometer for precise force measurement. It consists of two parts: a moving platform and a measurement stylus (Fig. 5.1). The platform includes a linear servo motor (MX80L, Parker, USA) capable of following a trapezoidal acceleration profile with a travel distance of 8 cm. The motor comes with a dedicated motor drive and a power supply (VIX250IH and XLPUS, Parker, USA). The stylus is made of stainless steel equipped with a six-axis force/torque sensor (Nano 17, ATI Industrial Automation, USA). A sensitive touch pen is attached to the lower end of the stylus to interact with the texture sample. A mass component at the top adjusts the total weight of the stylus

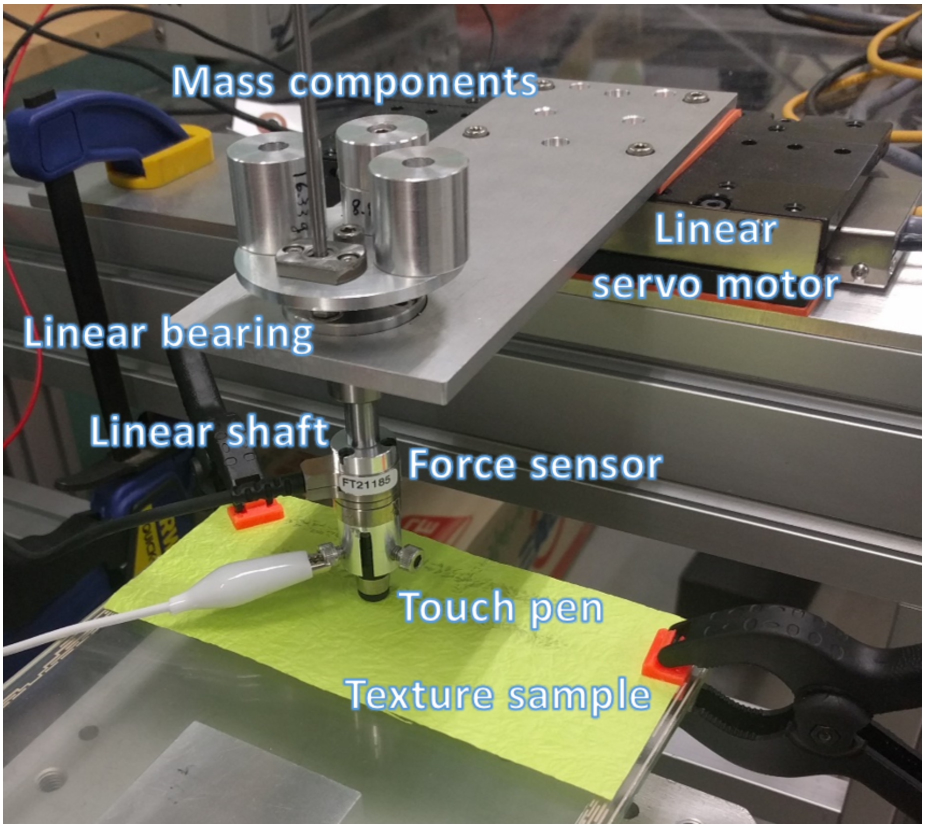


Figure 5.1: Motorized linear tribometer. The measurement stylus is attached to the moving platform using an Aluminum link.

and hence the normal pressure. The shaft is supported by a linear bearing and attached to the motor carriage using an aluminum link. The measurement stylus resembles a simple mechanical mass-damper system subject to Coulomb (dry) friction between the shaft and the linear bearing.

Each data collection trial starts with placing a texture sample on the mounting seat under the touch pen. A custom C++ program sends movement commands to the motor driver via RS232 and records the force data using a data acquisition card (PCI-6229, NI, USA) at 1 kHz rate. In this study, the data was taken under nine experimental conditions: three scanning speeds (3, 4, and 5 cm/s) and three vertical pressures (60, 85, and 110 g).

We apply a high-pass filter to the collected force data with a cut off frequency of 10 Hz to remove the different DC biases the force sensor encountered during moving left or right. We then manually extract the segments corresponding to the constant scanning speed from each data set. The segments are then concatenated to make a data stream for the texture sample under the given experimental condition.

5.1.2 Electrovibration Display

We use a capacitive touch panel (MicroTouch SCT3250, 3M, USA) to make an electrovibration display, as initially proposed in [4]. It consists of a transparent conductive layer coated with a thin insulator on top of a thick glass plate. Applying a high voltage waveform (around 100 Vpp) to the conductor and the electrically grounding human body modulates the friction between the sliding finger and the touch panel. We use a high performance piezo driver (MX200, PiezoDrive, Australia) that can amplify the input signal up to 200 Vpp. The panel also responds well to some touch pens; we tested several types and found one that creates almost the same level of electrostatic force as the human fingertip. The main reason using the touch pen is to eliminate the effects of the human skin properties that vary over time from force measurements, in order for consistent data collection.

5.2 Texture Model

In this section, we describe how to make an inverse neural network model of the input-output dynamics of the electrovibration display. We use this inverse model to synthesize actuation signals from the force recordings in order to create virtual textures on the display as similar as the real ones.

5.2.1 Inverse Dynamics Model

Our first goal is to develop a model that describes the dynamics relationship between the input actuation signal and the output frictional force of the electrovibration display. We follow a classic system identification procedure to estimate the system model based on observed input-output data. To achieve this, we 1) define input and output variables, 2) design proper actuation signals to drive the system, and 3) draw a dynamics model between the variables using an appropriate framework.

Input and Output

As for our electrovibration display, the input is actuation signal $u[n]$, and the output is lateral force $f_l[n]$. An usual approach is to build a model g such that $f_l[n] = g(u[n])$ and then use the inverse model g^{-1} to obtain $u[n]$ for desired $f_l[n]$ such that $u[n] = g^{-1}(f_l[n])$. Since an electrovibration display exhibits considerable nonlinearity in its behavior [13, 19, 31], nonlinear basis models such as neural networks are appropriate for system identification. The problem is that finding an inverse model for such nonlinear systems is not always feasible. An alternative is to identify the inverse model directly by using $f_l[n]$ as input and

$u[n]$ as output. This approach, however, requires care to obtain dense samples of $f_l[n]$, and we use neural networks for the inverse model identification.

Pseudo-Random Binary Signals

Next we need a proper actuation or perturbation signal to excite the system and observe its response. An open-loop experiment is required for such data collection, and the input signal must contain sufficient distinct frequencies in the frequency band of interest [66]. We use a Pseudo-Random Binary Signal (PRBS), which is a deterministic signal with white-noise like properties. PRBS is widely used for identification of linear systems [66] and occasionally for some nonlinear systems [67]. In case of nonlinear systems, its applicability depends on the nature of system nonlinearities. There are several successful implementations [68], as well as some failed attempts [69]. For example, a quadratic Wiener model can be identified using a PRBS while a first-order Hammerstein model cannot be. It is claimed that because the signal has only two levels, it may not excite certain nonlinearities of the system, so more input levels are required [70]. Nevertheless, we show that PRBS works in our case and can adequately capture the dynamic behavior of our electrovibration display.

A PRBS is characterized by three parameters: signal level $\pm c$, order n, and clock period B. A PRBS fluctuates between $-c$ and $+c$; its maximum period is $2^n - 1$; and it has to stay constant for at least B consecutive samples before it changes. To have a full-band signal that spans the entire frequency band up to the Nyquist frequency (sampling rate/2=500 Hz), clock period must be set to 1. Furthermore, considering the slowest scanning speed of 3 cm/s, it takes

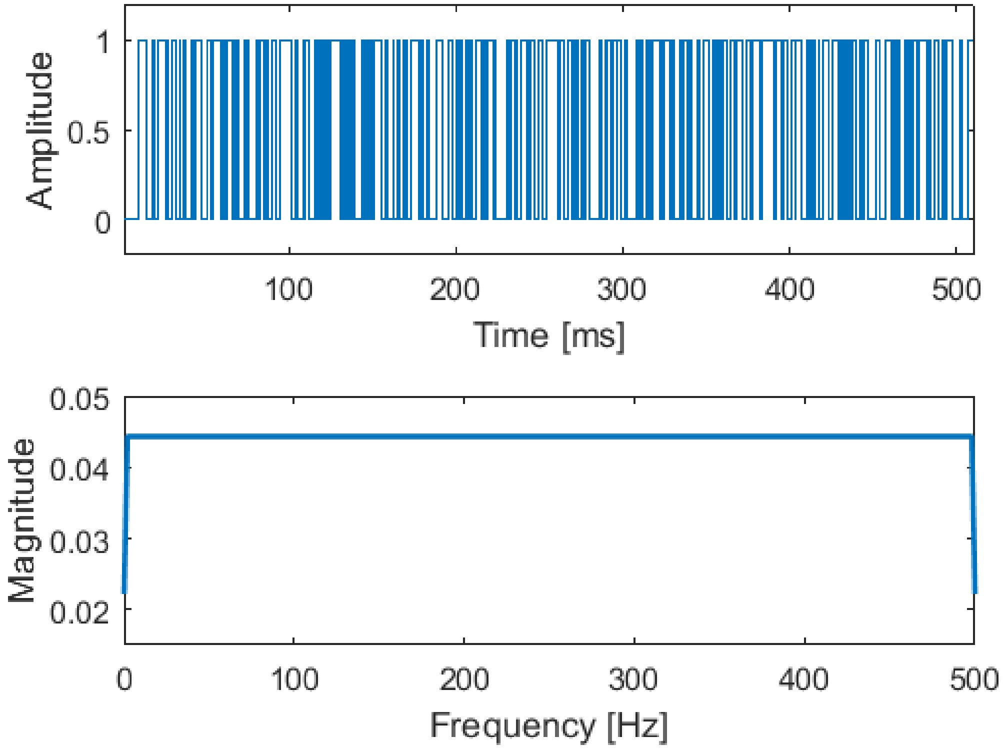


Figure 5.2: An example PRBS9 with $n = 9$.

approximately 2667 ms for our tribometer to complete an 8-cm travel. Therefore, the length of the PRBS should not be less than the longest travel time. For a signal with $n = 9$ (PRBS9 for short), the maximum length becomes $2^9 - 1 = 511$, and with six repetitions, we get a signal with the total length of 3066 ms. This is slightly longer than the maximum travel time. An example plot of PRBS9 with its frequency response is given in Fig. 5.2. It can be seen that PRBS9 has a constant frequency response in its entire frequency band. We use a Matlab function, `idinput()` to generate our desired PRBS.

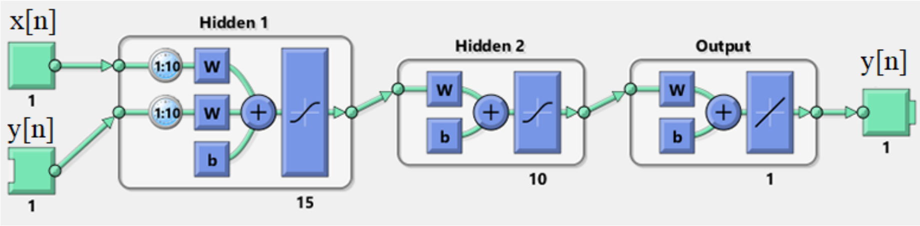
NARX Neural Network

With the PRBS in hand, we can now actuate the panel with it and record resulting lateral forces using our tribometer. In order to provide a proper framework to build a model among the observed variables, we use a NARX (Nonlinear AutoRegressive with eXternal input) neural network [71]. NARX neural networks are well suited for addressing nonlinear dynamics systems [67, 72, 73]. A NARX network has a close-loop structure with a feedback loop connecting the output to the input (Fig. 5.3(b)). The past values of both input and output are used to predict the future output. The defining equation for the NARX model is:

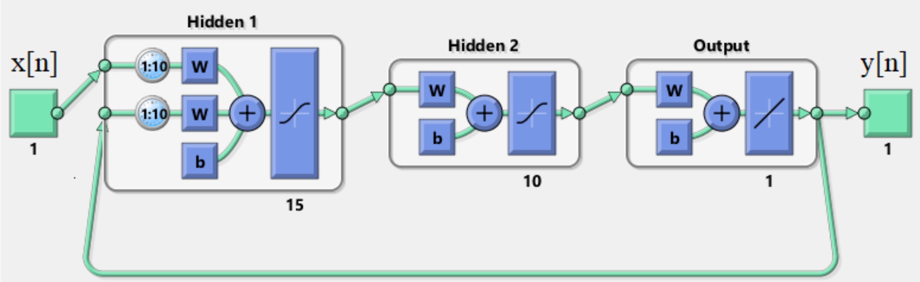
$$y[n] = f(y[n-1], ..., y[n-k_y], x[n-1], ..., x[n-k_x]), \tag{5.1}$$

where $y[n]$ is the output to be predicted and $x[n]$ is the external input exciting the system. k_y and k_x are the tap delays for feedback and external input, determining the numbers of past values involved in the estimation process.

The next step is to determine the number of hidden layers and the number of neurons in each layer, as well as the size of tap delays for each input sequence. We tested several different combinations and obtained nearly perfect performance with three layers that have 15, 10, and 5 neurons, respectively, and both tap delays of 10. We use the Matlab neural network toolbox for training and evaluation. Training is initiated by removing the feedback loop and inputting the desired target sequence along with the external input to the network (Fig. 5.3(a)). In our case, $x[n] = f_l[n]$ and $y[n] = u[n]$, where $u[n]$ is the designed PRBS9. For evaluation, we close the feedback loop by removing the reference PRBS9 from the input and instead feeding the estimates back (Fig. 5.3(b)). A result of close-



(a) Open-loop training.



(b) Close-loop evaluation.

Figure 5.3: Block diagram of the training and evaluation of NARX network. $y[n]$ is the reference signal (PRBS9) and $x[n]$ is the external input (recorded lateral force).

loop evaluation of the trained network is given in Fig. 5.4. The estimated PRBS (dashed red line) almost perfectly follows the reference PRBS (solid blue line).

5.2.2 Synthesizing Actuation Signals

For synthesis, we train inverse neural networks for different lateral scanning velocities and normal forces since the two variables affect texture responses to large extent [74, 75]. We obtain one neural network for each combination of scanning velocity and normal force. These inverse neural networks enable us to synthesize actuation signals for a texture material scanned under the same experimental condition.

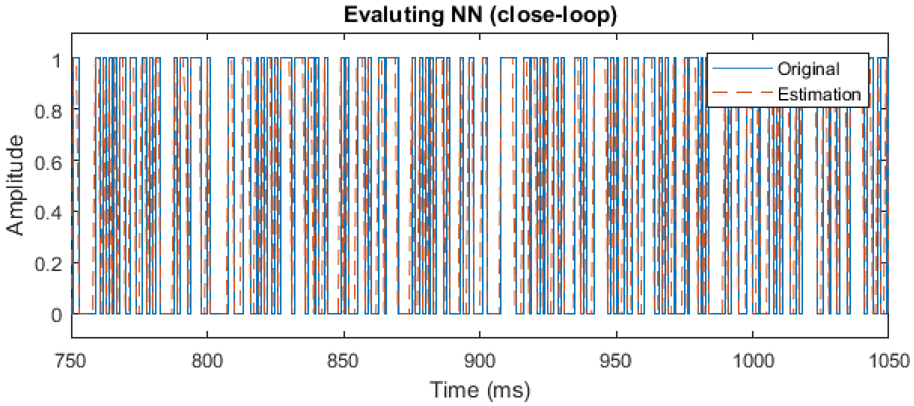


Figure 5.4: Example result of the training and evaluation of NARX network. For clarity, only 300 ms of the recordings are shown. The blue solid line is the reference PRBS9, and the dashed red line is an estimate from the neural network.

To this end, we first collect lateral forces from the surface of a real material using our tribometer under different scanning velocities and normal forces. We normalize each force sequence to handle the variability between readings from different materials. The same process is applied to the forces collected from the electrovibration display before training neural networks. Then we input each force reading to the corresponding trained neural network to generate a PRBS-like actuation signal. The estimated actuation signal attempts to recreate similar textural patterns to the real material once applied to the capacitive touch panel. However, these signals are only applicable to the same experimental conditions under which they were obtained. For arbitrary scanning velocity and normal force, we apply linear interpolation between the closest neighbors in the velocity-force grid similarly to [43, 55]. More details are provided in Section 5.3.3.

5.2.3 Error Metric

To assess the synthesis accuracy of virtual textures, we first visually compare the FFT plots of real and virtual textures, and then compute a relative spectral rms error as an error metric [43, 55]:

$$E_s = \frac{RMS(\mathcal{F}(f_v[n])) - \mathcal{F}(f_r[n])}{RMS(\mathcal{F}(f_r[n]))} \tag{5.2}$$

where $\mathcal{F}(\cdot)$ is the operator for fast Fourier transform and RMS is for computing the root mean square in the frequency domain. In addition, $f_v[n]$ denotes virtual forces and $f_r[n]$ real forces. We set the length of FFT equal to the sampling frequency, 1000 Hz, to generate a smooth and less spiky spectrum with the main components easy to distinguish. A smaller value of E_s indicates higher similarity between two spectra and hence higher similarity between virtual and real textures. We also applied the same error metric over repeated recordings of the same real material under the same experimental condition in order to establish a baseline (lower bound) of this metric.

5.3 Experimental Results

In this section, we evaluate the performance of our method of recreating virtual textures similar to the real ones. We present FFT plots for visual comparison and E_s values for objective evaluation. For that, we collected force data using the tribometer from six texture samples shown in Fig. 5.5. As shown in Fig. 5.6, we use four out of nine experimental conditions (blue squares) to train four neural networks and four others (red circles) for cross-validation.

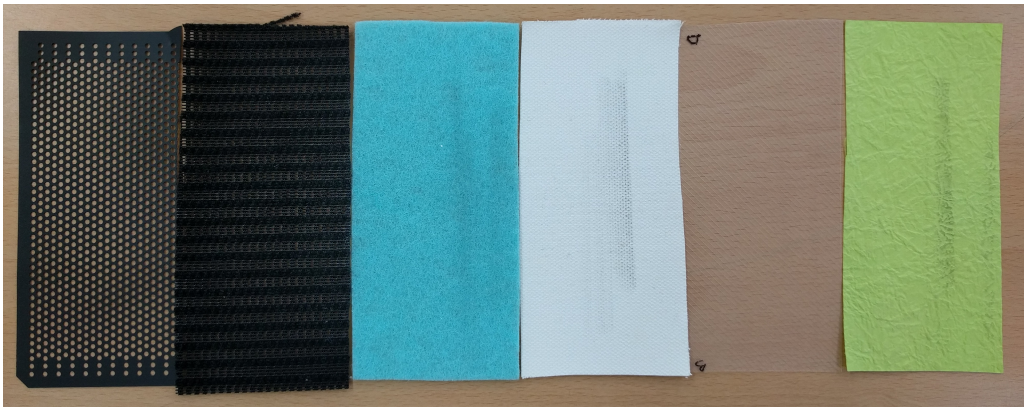
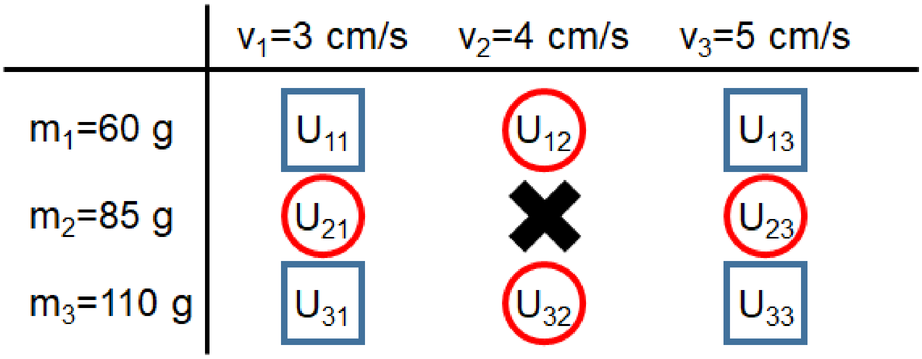


Figure 5.5: Six texture samples used in the evaluation. From left to right: dotted sheet (TS1), chair fabric (TS2), felt fabric (TS3), painting canvas (TS4), transparent plastic sheet (TS5), and scrunched paper (TS6).



	v_1=3 cm/s	v_2=4 cm/s	v_3=5 cm/s
m_1=60 g	U_{11}	U_{12}	U_{13}
m_2=85 g	U_{21}	✖	U_{23}
m_3=110 g	U_{31}	U_{32}	U_{33}

Figure 5.6: Force-velocity grid for cross-validation. m_i indicates mass and v_j scanning velocity. For experimental conditions highlighted by blue squares, the actuation signal is obtained from the corresponding trained neural networks. For others marked by red circles, the signals are obtained by linearly interpolating neighbors.

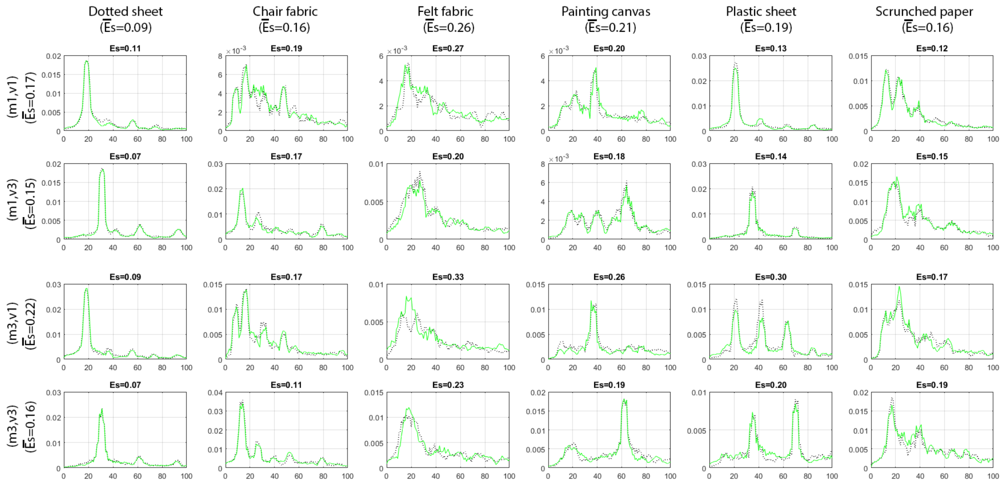


Figure 5.7: Baseline FFT plots with error metrics. Dotted black line is the first repetition, and solid green line is the second repetition. The average errors are given inside parentheses on the top and left side.

5.3.1 Baseline Performance

We first evaluated the similarity between the repeated measurements from the same material under the same experimental condition to establish a baseline for our error metric E_s. Ideally, E_s should be zero in such cases, but due to various noises and uncertainties, we expect the values of E_s greater than zero. The baseline FFT plots for the six materials are given in Fig. 5.7 with the E_s values. Average E_s values are shown on top of the columns for each material, and on the left of the rows for each experimental condition.

Several observations can be made from Fig. 5.7. Visually, the repeated measurements look very similar, and for some materials such as dotted sheet, they are almost identical. The smallest E_s is 0.07 (dotted sheet), and the largest is 0.33

(felt fabric). On average, dotted sheet showed the highest consistency between repeated readings ($\bar{E}_s = 0.09$), while felt fabric showed the lowest ($\bar{E}_s = 0.26$). Among the experimental conditions, (m_1, v_3) showed the highest consistency ($\bar{E}_s = 0.15$), while (m_3, v_1) showed the lowest ($\bar{E}_s = 0.22$).

We can also observe the effects of increasing scanning speed and vertical mass (normal force) on the spectra. Increasing the scanning speed tended to shift the main frequency component to right (to higher frequencies). This can be seen by comparing the first and second rows and the third and fourth rows. However, with heavier mass, only the level of power density increased, which can be confirmed by comparing the first and third rows and the second and fourth rows. Furthermore, for dotted sheet the spectrum shapes were almost the same in all the conditions. For chair fabric, the shape changed with scanning speed but not with mass. For plastic sheet, additional main components appeared with heavier mass. It seems that the minor components in m_1 are emphasized in m_3.

5.3.2 Training Performance

Next, we examine the similarity between virtual forces generated by synthesized PRBS-like actuation signals and real texture forces. Initial examinations showed that when the actuation signal is amplified with the default gain (equal 1.0 correspond to 100 Vpp), in some cases, the levels of virtual power densities do not match those of real ones. Thus we tested a range of gains from 0.6 to 1.4 with a step size of 0.2, and picked one that generated almost the same maximum magnitude of power density (Table 5.1).

The resulting FFT plots and E_s values are shown in Fig. 5.8. In most cases,

Table 5.1: Adjusted gains for each material (TS1–TS6) under each experimental condition.

Exp. Cnd.	TS1	TS2	TS3	TS4	TS5	TS6
(m_1, v_1)	1.4	1.0	0.6	0.6	1.4	1.2
(m_1, v_3)	1.0	1.4	1.0	0.6	1.0	1.0
(m_3, v_1)	1.0	1.0	0.6	0.6	0.6	0.6
(m_3, v_3)	1.0	1.4	1.0	0.8	0.6	1.0

the neural networks estimated actuation signals successfully so that the virtual spectrums were similar to the real ones. Note that the E_s values in Fig. 5.8 are absolute ones, and they should be interpreted in reference to the baseline E_s values in Fig. 5.7.[1]

Among the six materials, painting canvas showed the highest similarity ($\bar{E_s} = 0.29$), while chair fabric showed the lowest ($\bar{E_s} = 0.49$). Plastic and dotted sheet showed the second best performance with $\bar{E_s} = 0.37$, followed by felt fabric and scrunched paper with $\bar{E_s} = 0.44$. A closer look reveals that the three best materials have some sort of uniform patterns on their surfaces, are made from sturdy materials, and generally feel rough. In contrast, the three worst materials are all made from soft fabrics with random or no clear textural patterns on their surfaces. As for the experimental condition, although (m_1, v_3) showed the lowest average error, error levels were similar regardless of scanning velocity or normal force.

---

[1] It is unclear how to compare two different E_s values, e.g., it can be additive or multiplicative, or neither.

5.3.3 Cross-validation

We next show how well for an untested condition (marked by red circles in Fig. 5.6) a linear interpolation between adjacent tested conditions (marked by blue squares), generates an actuation signal. For example, U_{12} is obtained by taking a weighted average between U_{11} and U_{13} with the weights equal to the adjusted gains. Similarly U_{21} is obtained by taking a weighted average between U_{11} and U_{31}. The FFT plots along with E_s values are given in Fig. 5.9.

An immediate observation reveals that taking average has almost worked for the conditions with the same scanning speed but with different mass; check the second row for (m_2, v_1) and the third row for (m_2, v_3). Most of the main components are preserved buy with some variations in the power densities. However, in the conditions having the same mass, (m_1, v_2) in the first row and (m_3, v_2) in the forth row, the obtained spectra are clearly different from the references. It seems the main components of lower and higher speeds appeared in the resulting spectra. This can be clearly observed for dotted and plastic sheet. This behavior can be explained based on the fact that the Fourier transform of a linear combination of two time-domain signals will include the main components of both spectra. This is evidently the weakness of applied interpolation method. This effect is less noticeable for materials with random textural patterns such as felt fabric and scrunched paper.

We tested a smaller interval and performed interpolation between (m_3, v_1) and (m_3, v_2) to obtain an actuation signal for $(m_3, 3.5\,cm/s)$. From spectral plots (not shown here), we could still observe the appearance of both components in

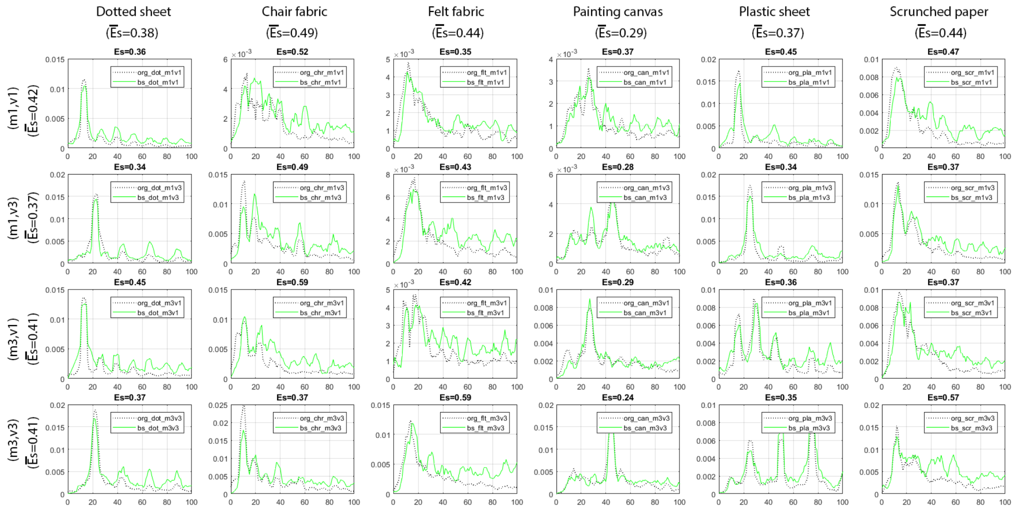


Figure 5.8: FFT plots comparing virtual and real textural forces. Dotted black lines indicate real forces, and solid green lines virtual ones. Average errors are given inside the parentheses on the top and left sides.

the resulting spectra but to a smaller extent; the average error $\bar{E}_s$ decreased from 0.70 to 0.65. It seems with smaller velocity intervals, the accuracy of this linear interpolation method increases. While reducing velocity intervals can be a temporary solution, a nonlinear averaging or interpolation scheme would be a better and ultimate solution.

The best performance is achieved for scrunched paper with $\bar{E}_s = 0.49$ and worst for dotted sheet with $\bar{E}_s = 0.78$. Among conditions, (m_2, v_1) showed lowest error ($\bar{E}_s = 0.55$) and (m_3, v_2) showed highest ($\bar{E}_s = 0.70$).

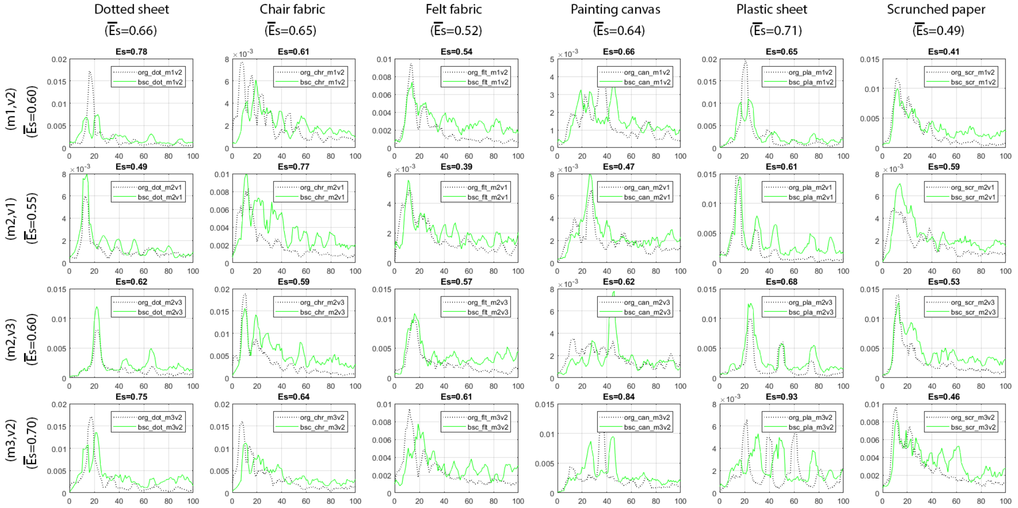


Figure 5.9: FFT plots for cross-validation comparing virtual and real textural forces for the experimental conditions not used for neural network training. Dotted black lines indicate real forces, and solid green lines virtual ones. Average errors are given inside the parentheses on the top and left sides.

5.4 Conclusions

In this study, we proposed an inverse NARX neural network model to generate PRBS-like actuation signals to mimic real textures on an electrovibration display. The networks are trained based on the lateral forces collected from the surface of display as a result of applying a full-band PRBS. Once trained, the networks are used to estimate an actuation signal from lateral forces collected from real samples. By comparing the spectra in frequency domain, we showed promising results recreating virtual textures similar to the real ones. We also showed preliminary cross-validation results to estimate an actuation signal for untested experimental conditions by taking average between adjacent tested conditions. This simple approach showed acceptable results for cases with the same scanning velocity but not for cases with the same mass.

For future work, we intend to conduct a human user study to evaluate the perceptual quality of the textures generated by our approach. In addition, to better address the nonlinear nature of electrovibration display, we have plans to use m-level pseudo-random signals (PRMS) instead of 2-level or binary ones (PRBS) to perturb the display and identify its dynamic system. Finally, we believe a more sophisticated interpolation scenario can result in better estimation of actuation signals for untested conditions.

Chapter VI.
Improved Interpolation Scheme

In the previous chapter, we introduced a basic interpolation scheme to estimate and generate actuation signals for untested experimental conditions, the ones not being used for training neural networks. We showed that while it works appropriately for the conditions with the same scanning velocity but different masses, it does not work well for the conditions with the same mass but different scanning velocities. The biggest observed issue was that the main frequency components from the lower and higher velocities both appear in the forces resulted from the interpolated (=estimated) signals (Fig. 5.9).

In this chapter, we propose a solution to solve this problem. We first redefine the problem with some basic examples and investigate its origin. Then we explain our algorithm as an effective way to fix the problem accompanied by several examples. Then we show how this algorithm can be applied as an improved interpolation scheme to our texture rendering method. At the end, we conduct a human user study to rate similarity between real and virtual textures produced by our method to show the effectiveness of the proposed algorithm.

6.1 Problem definition

In the previous chapter, we showed the effect of scanning velocity on the location of main frequency component of the collected forces from the surface of real materials. In most cases, with the increased scanning velocity, the component shifted right (to higher frequencies) in the frequency domain. This behavior is captured by the neural networks we trained over the collected data. Therefore, for the conditions with different scanning velocity, the neural networks have learned such a behavior and the estimated actuation signals have inherited such an attribute. This means, for a given texture sample, the actuation signal estimated under the slower velocity, produces forces with lower frequency components. In a similar way, the actuation signal estimated under the faster velocity, generates forces with higher frequency components. As such, when these two signals are linearly combined (by taking weighted average), the generated signal aims to recreate both lower and higher frequency components when FFT is applied on the resulting forces collected from the 3M panel. This behavior can be explained based on the fact that FFT is a linear operator which a combination of signals with different frequencies will result in a combination of their corresponding power spectra after FFT being applied.

We provide a basic example to show this. Let us assume we have two sine waves, $y_1 = \sin(2\pi f_1 t)$ and $y_2 = \sin(2\pi f_2 t)$, with two distinct frequencies f_1

and f_2. Applying FFT, each sine wave creates a power spectrum with the main component located in f_1 or f_2 (Fig. 6.1). Our intention is to generate a sine wave y_3 for an arbitrary frequency f_3 in which $f_1 \leq f_3 \leq f_2$. A weighted averaging scheme yields to:

$$y_3 = \mathcal{WA}(y_1, y_2, f_1, f_2, f_3) = \frac{f_2 - f_3}{f_2 - f_1} y_1 + \frac{f_3 - f_1}{f_2 - f_1} y_2. \tag{6.1}$$

where $\mathcal{WA}$ is an operator for taking weighted average between two signals. In extreme case, if $f_3 = f_1$ then $y_3 = y_1$ and if $f_3 = f_2$ then $y_3 = y_2$. For a middle frequency, $f_3 = (f_1 + f_2)/2 = 15$ Hz, the formula yields to $y_3 = 0.5y_1 + 0.5y_2$. As it can be seen from Fig. 6.1, the outcome is not satisfactory because we expected a sine wave at frequency of $f_3 = 15$ Hz with full power density but instead we obtained two sine waves at frequencies $f_1 = 10$ Hz and $f_2 = 20$ Hz with half power density. This is the same problem we observed with our basic interpolation scheme applied on actuation signals with different scanning velocities. We do not want to have two separate components at two distinct frequencies, we want to have one single component at the intended frequency. Any linear interpolation (averaging) scheme will yield to such an outcome. Therefore, either a nonlinear or a modified linear scheme is required to achieve the desired output. We explain our solution in the next section.

6.2 Potential solution

A closer look into the problem of having multiple components in the outcome reveals a potential solution too. Referring to the given example of two sine waves, if we shift the lower component to the right and the higher component to the left

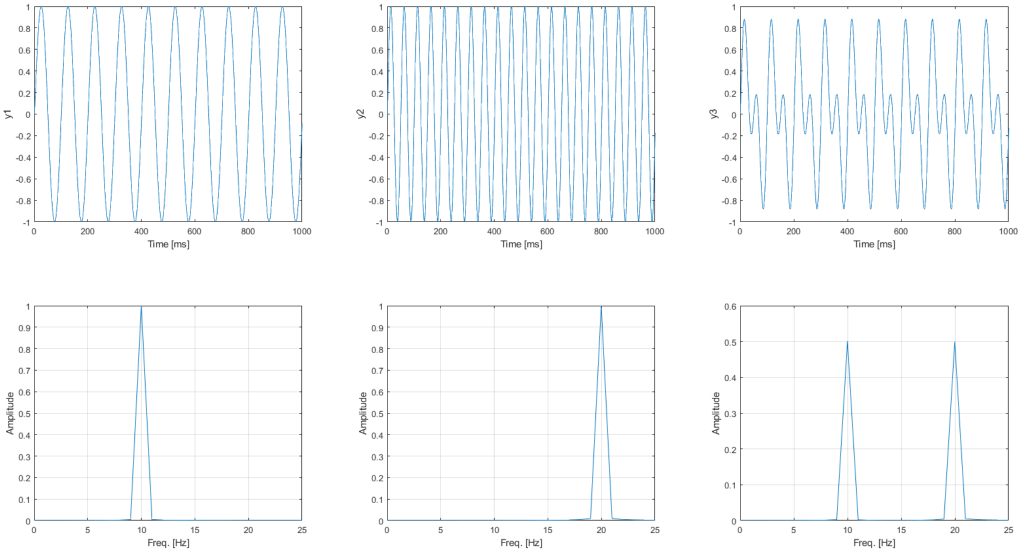


Figure 6.1: Example of two sine waves with frequencies $f_1 = 10$ Hz and $f_2 = 20$ Hz, and their corresponding FFT plots. $y_3 = 0.5y_1 + 0.5y_2$ is calculated as an estimation for $f_3 = 15$ Hz by taking weighted average from y_1 and y_2. Its time and frequency domain plots are given on the right.

in the frequency domain, then taking average between the shifted signals will result in a component at the desired frequency and appropriate power density. This is a simple yet effective technique comparing to a complicated and possibly hard to implement nonlinear alternative.

To shift a component in the frequency domain, we require to perform a re-sampling process, either down-sampling or up-sampling, in the time domain. While down-sampling is also called *decimation* in signal processing, but with an appropriate *decimation factor*, d.f., we can perform both re-sampling processes. d.f. determines the degree of down-sampling if it is greater than 1 and up-sampling if it is less than 1. Decimation with $d.f. \geq 1$ yields a signal with its frequency component shifted right (to higher frequency) and with $d.f. \leq 1$ to a signal with its frequency component shifted left (to lower frequency).

Let us first define an operator, $\mathcal{RS}(.)$, to perform re-sampling: $y^r = \mathcal{RS}(y, p, q)$. y^r is the re-sampled version of y with a rational decimation factor $d.f. = p/q$. It technically means that y is being re-sampled at p/q times the original sample rate. p and q are obtained from the current and intended frequencies. For example, to shift right a component from f_1 to f_3, decimation factor should be $d.f. = f_1/f_3 \leq 1$ and vice versa to shift left from f_2 to f_3, $d.f. = f_2/f_3 \geq 1$. Next we apply weighted average on the resulting two re-sampled signals. Let us denote the re-sampled version of y_1 by $y_1^r = \mathcal{RS}(y_1, f_1, f_3)$ and y_2 by $y_2^r = \mathcal{RS}(y_2, f_2, f_3)$. Then y_3, an estimation based on y_1^r and y_2^r is obtained from:

$$y_3 = \mathcal{WA}(y_1^r, y_2^r, f_1, f_2, f_3) = \frac{f_2 - f_3}{f_2 - f_1} y_1^r + \frac{f_3 - f_1}{f_2 - f_1} y_2^r. \tag{6.2}$$

This process is simulated for the same sine waves used in the previous example. The intended frequency is $f_3 = 15$ Hz, so we require to shift right the component at $f_1 = 10$ Hz to $f_3 = 15$ Hz and shift left the component at $f_2 = 20$ Hz to $f_3 = 15$ Hz. The time and frequency domain plots are given in Fig. 6.2. As it can be seen, each component is first shifted to the desired frequency then combined to generate a single component with full power density.

Another example is provided for two sine waves with different frequencies and different amplitudes: $y_1 = A_1 \sin(2\pi f_1 t)$ and $y_2 = A_2 \sin(2\pi f_2 t)$ with $A_1 = 1.0$ and $f_1 = 10$ Hz and $A_2 = 0.4$ and $f_2 = 20$ Hz. Our intention is to generate a signal at $f_3 = 19$ Hz. The time and frequency domain plots are given in Fig. 6.3.

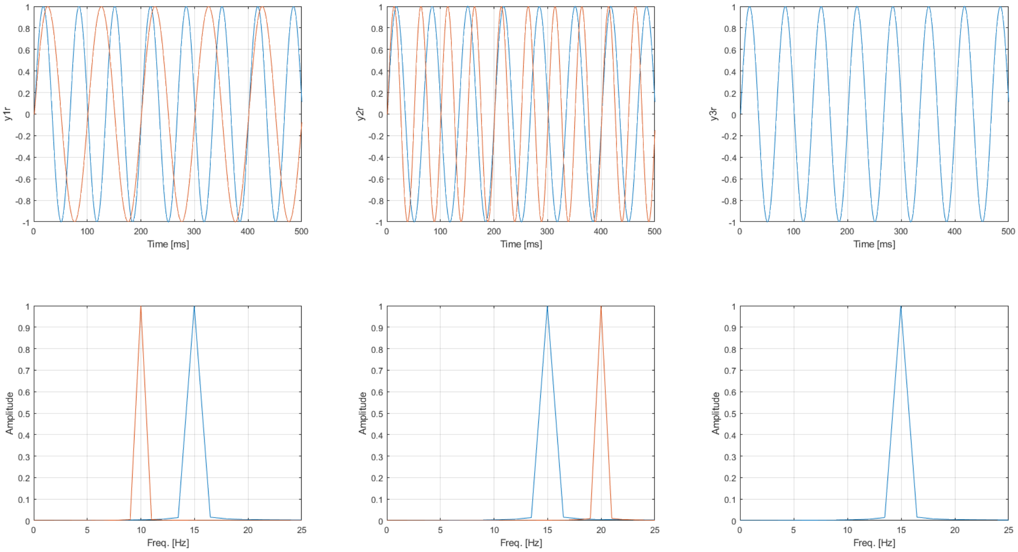


Figure 6.2: Example of re-sampling two sine waves with frequencies $f_1 = 10$ Hz and $f_2 = 20$ Hz to generate a signal (sine wave) at the desired frequency of $f_3 = 15$ Hz. Time domain plots are given on the top and FFT plots on the bottom. The red plots are the original signals and the blue ones are re-sampled ones. y_1^r, blue plot, is the re-sampled version of y_1 and y_2^r is of y_2. y_3 is obtained by linearly combining y_1^r and y_2^r: $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 15) = 0.5y_1^r + 0.5y_2^r$.

As it can be seen the interpolated signal y_3 is more similar to y_2 than y_1 in terms of power density because the intended frequency, f_3, is closer to the f_2 than f_1.

We finish this section with a final example. This time we intend to obtain an estimation from $y_1 = 0.3\sin(2\pi 10t)$ and $y_2 = \sin(2\pi 20t)$ at frequency $f_3 = 12$ Hz. A single plot illustrating the re-sampling and averaging process is shown in Fig. 6.4. Again the outcome is more similar to y_1 in terms of power density because $f_3 = 12$ Hz is closer to $f_1 = 10$ Hz.

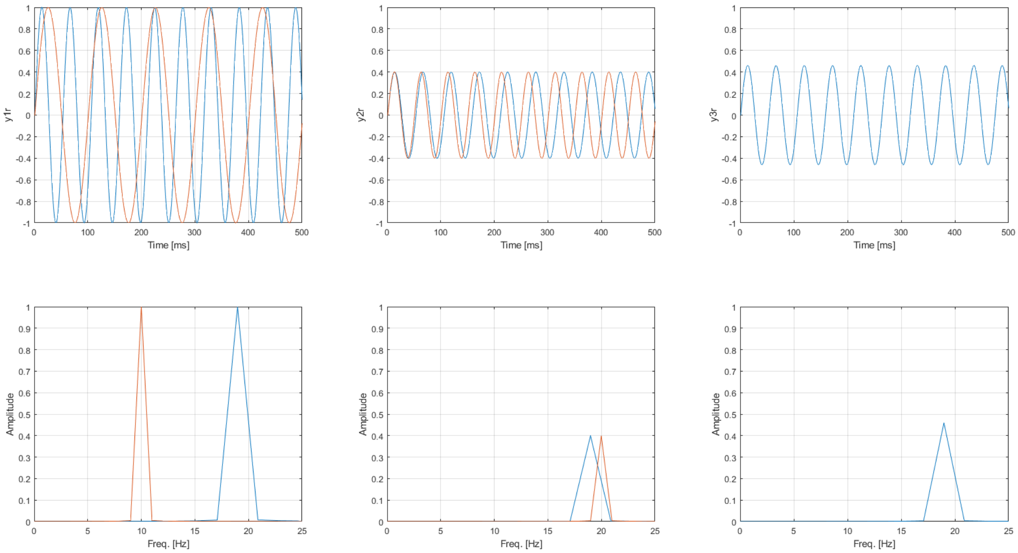


Figure 6.3: Another example of re-sampling and combining two sine waves with different frequencies and different amplitudes. For the desired frequency of $f_3 =$ 19 Hz the signal is obtained from $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 19) = 0.1y_1^r + 0.9y_2^r$.

6.3 Test on real texture samples

A key point applying the aforementioned re-sampling process is to know the desired frequency we intend to land and also the lower and higher frequencies we intend to shift. For the given examples, we knew all these frequencies in advance, but how about for a real texture sample? We are basically seeking to apply such an algorithm on the estimated actuation signals obtained from the trained neural networks. Just to recall, we train a neural network under a known normal pressure and scanning velocity. For the untested conditions, we intend to interpolate signals from the adjacent tested conditions. When the two neighborhood conditions have different scanning velocities we must apply the

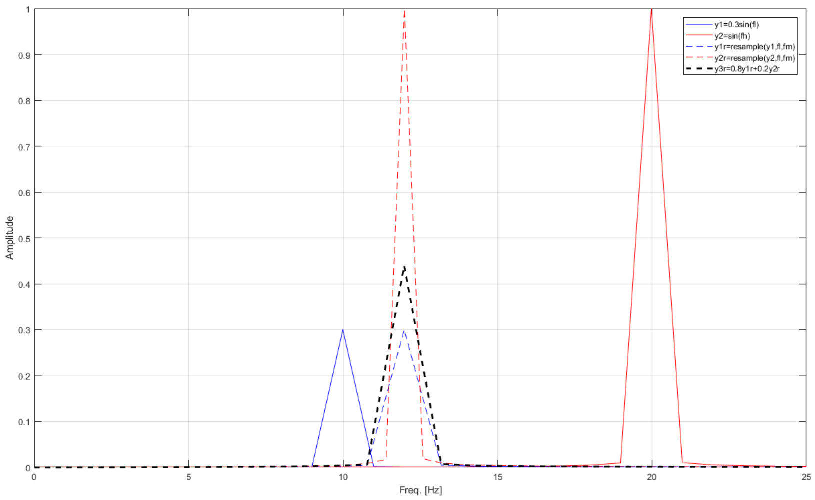


Figure 6.4: Another example of re-sampling and combining two sine waves with different frequencies and different amplitudes. For the desired frequency of $f_3 =$ 12 Hz the signal is obtained from $y_3 = \mathcal{WA}(y_1^r, y_2^r, 10, 20, 12) = 0.8y_1^r + 0.2y_2^r$.

re-sampling process, otherwise, when they have the same scanning velocity but different normal pressures we do not need to do so. For an untested condition, we just know the lower and higher velocities and the current scanning velocity user is applying (we will discuss this in the human user study section). But we still need to know the corresponding frequencies of the main component to perform the re-sampling process. For that, we require to investigate the relationship between the scanning velocity and the frequency of the main component for a given texture sample.

As an initial test, we investigate such a behavior for a texture sample with an almost perfect sinusoidal patterns on its surface (Fig. 6.5). The reason is that, we expect to see a single powerful component in the forces collected from this sinusoidal surface, making it an ideal candidate to study the effect of scanning velocity on the frequency of the main component.

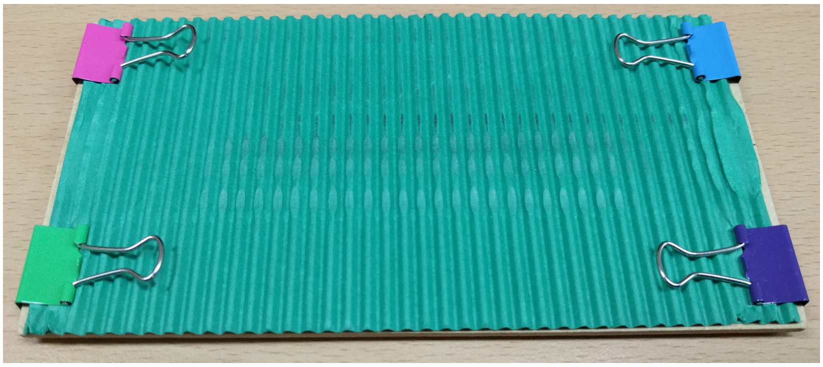


Figure 6.5: Rippled paper with almost perfect sinusoidal surface patterns.

We used our tribometer to collect lateral forces from the surface of rippled paper with five different scanning velocities: 3,4,5,6, and 7 cm/s. A single plot showing all five FFT plots is given in Fig. 6.6. The plot is annotated with the values of the corresponding main component. We then plotted these frequencies with respect to their corresponding scanning velocities (the inset plot in the Fig. 6.6). As it can be seen, there is a perfect linear relationship between frequency and velocity. This is an important finding, because frequencies being linearly related to scanning velocities, makes it possible to replace frequencies with velocities in the re-sampling and interpolation process. This means, we do not need to know the lower and higher frequencies and also the intended frequency to apply the re-sampling and then interpolation, we instead can use the lower and higher velocities and the current user applied scanning velocity for those purposes. We have investigated the same relationship for all six materials used in the previous chapter. For each material, we found a linear relationship between the frequency of the main component and the corresponding scanning velocity. The details are provided in appendix A. Realizing a linear relation between frequency and velocity, we can replace frequency with velocity in Eqn. 6.2 and re-define the interpolation averaging scheme accordingly:

$$\begin{aligned} y_3 &= \mathcal{WA}(y_1^r, y_2^r, f_1, f_2, f_3) \\ &= \frac{f_2 - f_3}{f_2 - f_1} y_1^r + \frac{f_3 - f_1}{f_2 - f_1} y_2^r \\ &= \frac{v_2 - v_3}{v_2 - v_1} y_1^r + \frac{v_3 - v_1}{v_2 - v_1} y_2^r \\ &= \mathcal{WA}(y_1^r, y_2^r, v_1, v_2, v_3). \end{aligned} \tag{6.3}$$

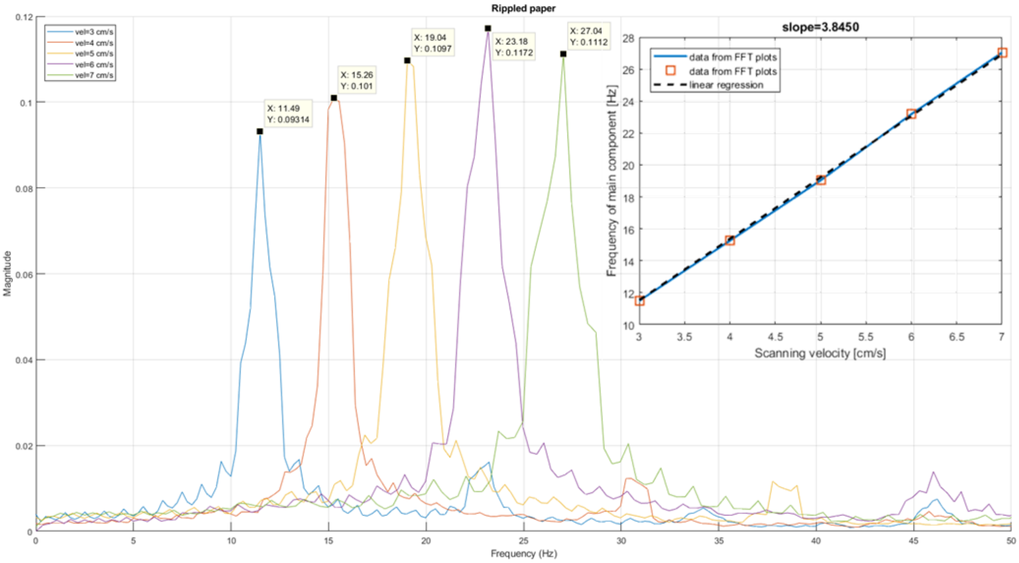


Figure 6.6: Power spectra of forces collected from the surface of rippled paper under five different scanning velocities. The inset plot at the upper right corner, shows a linear relationship between the frequencies of the main component and their corresponding velocities.

6.4 Improved interpolation scheme

With the improved interpolation algorithm for the conditions with different scanning velocities, we are now being able to define a complete interpolation scheme for a general untested condition. Just to recall, we train neural networks based on forces collected under known normal pressures and scanning velocities, one network for each condition. The neural network is used to generate a PRBS-like actuation signal under the same condition. For other untested conditions, the ones are not used to train the neural networks, we propose to interpolate an actuation signal from the four adjacent nodes in pressure-velocity grid (Fig. 6.7). Each node, $U_{i,j}$, represents an estimated actuation signal obtained from the trained neural network under the given normal pressure and scanning velocity, (m_i, v_j). The interpolation scheme has two parts. First, we obtain a signal from the nodes with the same scanning velocity but different masses. Second, we properly re-sample the newly generated signals and then take weighted average between them.

By assuming that user applied pressure is m_u, $(m_i \leq m_u \leq m_{i+1})$, and user applied scanning velocity is v_u, $(v_j \leq v_u \leq v_{j+1})$, the whole interpolation process can be formulated as below:

$$
\begin{aligned}
&1.1. \quad U_{u,j} = \mathcal{WA}(U_{i,j}, U_{i+1,j}, m_i, m_{i+1}, m_u)\\
&1.2. \quad U_{u,j+1} = \mathcal{WA}(U_{i,j+1}, U_{i+1,j+1}, m_i, m_{i+1}, m_u)\\
&2.1. \quad U^r_{u,j} = \mathcal{RS}(U_{u,j}, v_j, v_u)\\
&2.2. \quad U^r_{u,j+1} = \mathcal{RS}(U_{u,j+1}, v_{j+1}, v_u)\\
&2.3. \quad U_{u,u} = \mathcal{WA}(U^r_{u,j}, U^r_{u,j+1}, v_j, v_{j+1}, v_u)
\end{aligned}
\tag{6.4}
$$

The sub-steps 1.1 and 1.2 are about taking weighted average between signals with same scanning velocity, sub-steps 2.1 and 2.2 are about re-sampling the outcomes, and sub-step 2.3 is about taking average between the two re-sampled signals. For the user conditions with velocities slower or faster than the lower and upper velocity limits, or masses lighter or heavier than the lower and upper mass limits, only the extreme limits are considered. More specifically, for user velocities outside $[v_{min}, v_{max}]$, or masses outside $[m_{min}, m_{max}]$, v_u or m_u is set to the corresponding extreme. For example, if $v_u \leq v_{min}$, then we assume $v_u = v_{min}$. Similarly, if $m_u \geq m_{max}$, then we set $m_u = m_{max}$. This is equivalent to ignoring second interpolation step for out-of-range user velocities and ignoring first step for out-of-range user pressures.

We tested this interpolation scheme on the rippled paper. We trained neural networks for the conditions with two normal pressures (60 and 110 g) and five scanning velocities (3, 4, 5, 6, and 7 cm/s). The cross-validation is done for the middle conditions with one pressure (85 g) and four velocities (3.5, 4.5, 5.5, and 6.5 cm/s). The results are given in Fig. 6.8. In all cases, a decent agreement between the two spectra, real vs. virtual (=rendered), can be seen, proving the

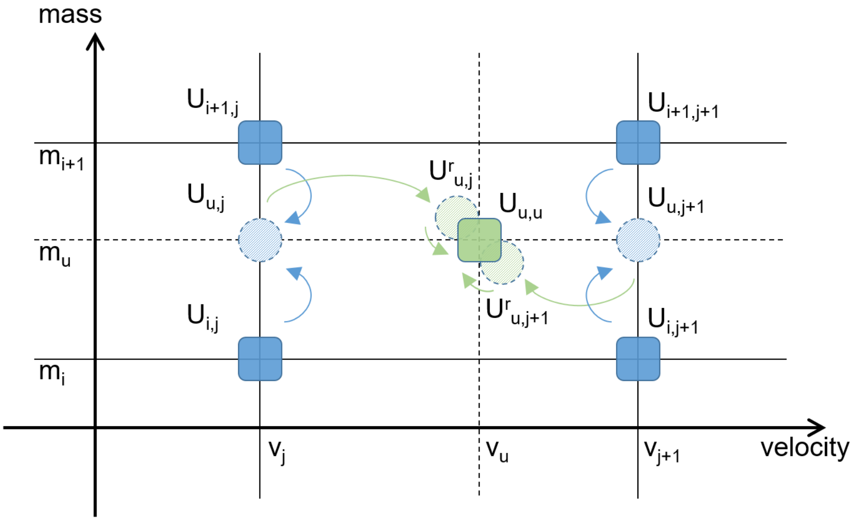


Figure 6.7: Improved interpolation scheme. (m_u, v_u) denotes user applied pressure and velocity. Blue squares are the adjacent actuation signals obtained from neural networks trained under corresponding experimental conditions. Blue arrows indicate sup-steps 1.1 and 1.2, applied on the vertical nodes to yield blue dashed circles. Long green arrows indicate sub-steps 2.1 and 2.2, applied on the horizontal nodes to yield green dashed circles. Short green arrows, indicate the final sub-step 2.3, applied on the re-sampled signals (green dashed circles) to obtain green square, the output of interpolation.

effectiveness of the proposed interpolation scheme.

6.5 Human user study

So far, we have shown that our texture modeling can adequately create virtual spectra similar to the real spectra for both tested and untested conditions. We evaluate the performance of our texture modeling and rendering algorithm by means of a user study. The emphasis is on the perceptual similarity between

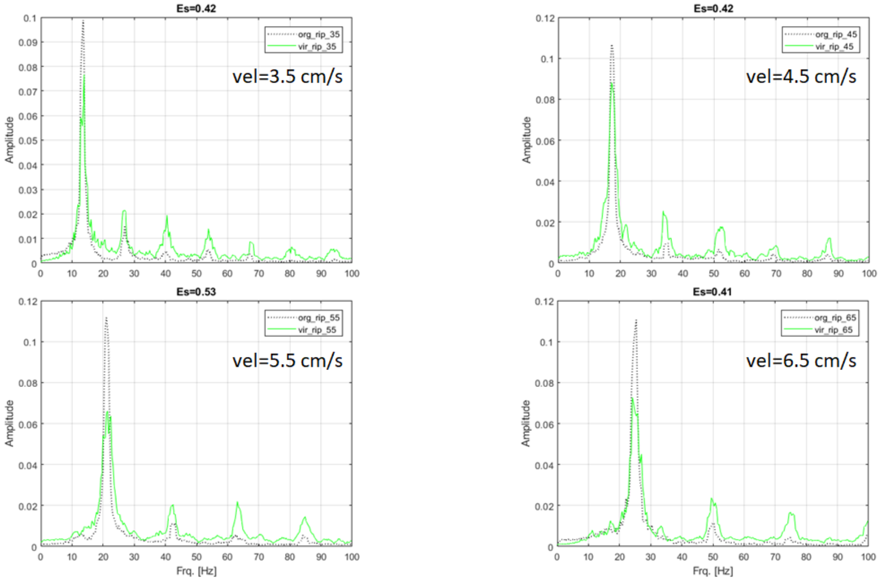


Figure 6.8: Real vs virtual spectra for rippled paper. The cross-validation conditions include one normal pressure (85 g) and four scanning velocities (3.5, 4.5, 5.5, and 6.5 cm/s). Recordings from rippled paper are shown with dotted black lines and recordings from 3M panel actuated with the interpolated signals with solid green lines. In all four conditions, a satisfactory agreement between the two spectra, real vs. virtual, can be seen, proving the effectiveness of the proposed interpolation scheme. The error metric values, E_s, are shown on top of each plot.

the virtual and the real textures. This user study is approved by the Institutional Review Board at the author's institution (PIRB-2017-E070).

6.5.1 Methods

Participants

We recruited twenty participants (13 males and 7 females; 18–26 years old with an average of 22.4) who had no prior experiences on using variable friction displays including electrovibration. None of them reported any known sensory or motor impairments. Participants signed an informed consent form after we explained the goals and procedure of the experiment. Each participant was paid KRW 5,000 (approximately USD 4.5) after the experiment.

Texture materials

We applied the modeling and rendering methods on six texture materials used in the previous chapter:

- Dotted sheet: made of a thin plastic sheet with holes all over its surface
- Chair fabric: made of a soft material with woven patterns
- Felt fabric: made of a thick soft material with no distinguishable patterns
- Painting canvas: made of a rough and sturdy linen with uniform nodes
- Plastic sheet: made of a rigid plastic sheet with diagonal ridges
- Scrunched paper: made of a thick paper with random textural patterns

They are chosen on the basis to represent different surface texture. Chair and felt fabrics feel soft while others generally feel rough.

Performance measure

Given a pair of textures, either real vs. real or real vs. virtual, participants were asked to evaluate their overall similarity on a scale of 0–100, with 0 being completely different and 100 being perfectly similar. The participants were told to focus on the surface textural patterns not the materials themselves.

Task and procedure

During the experiment, participants sat in a chair in front of a desk on which the experimental setup was placed (Fig. 6.9). The setup included a 3M panel, as electrovibration display, equipped with an IR-frame (T Series 10”, E&T Tech, China [76]) on the top to detect finger position and a bar type load-cell (CBCL-2L, CAS Scale, South Korea [77]) on the bottom to measure user applied pressure. Next to the panel and on the right side is a seat to place the real texture sample. We tried to remove all irrelevant sensory cues, therefore participants' vision to the 3M panel and texture sample was blocked by a curtain.

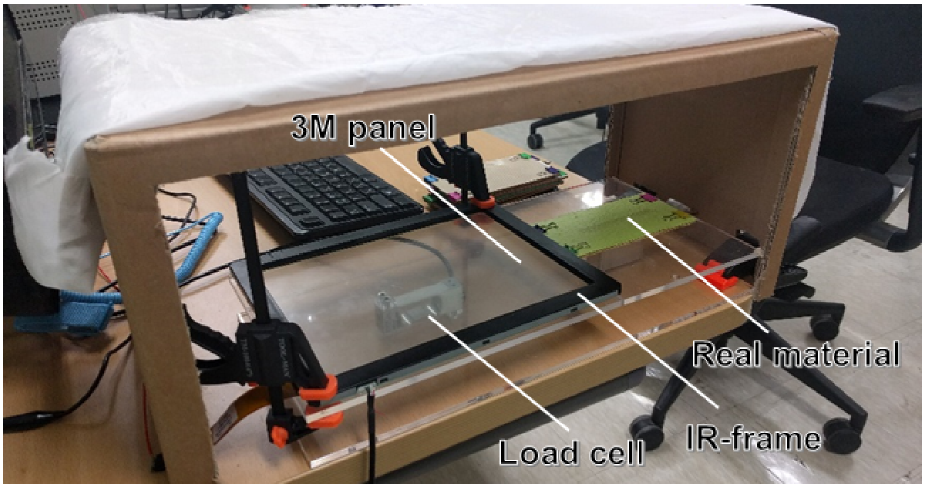


Figure 6.9: Experimental setup used for human user study. 3M capacitive touchscreen is used as electrovibration display. Under the panel a load cell is installed to measure user's pressure and on top of it an IR-frame to track user's finger position.

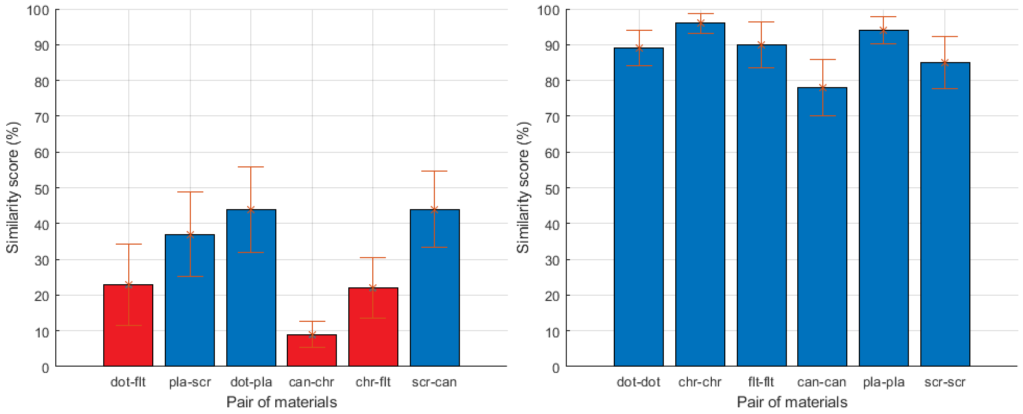
The experiment consisted of two phases. In the first phase, participants were provided with pairs of only real texture samples and were asked to rate their similarity. One sample is placed on the material seat to the right of user and another sample on the 3M panel to the left of user. The objective was to measure the upper and lower bounds of the overall similarity since humans tend to avoid giving extreme values (e.g., see [78]). Additionally, this phase helped participants stabilize their perceptual bases and scales for similarity scores. The participants were randomly presented with 12 pairs of materials, six of which with the same materials and six with the different materials representing distinguishable texture sensation. Among six pairs of different materials, three pairs were intentionally chosen from the materials having almost similar feeling (`pla-scr`, `dot-pla`, and `scr-can`) and three pairs from the materials having different feelings (`dot-flt`, `can-chr`, and `chr-flt`). The former is called *moderately-different* group and the latter *highly-different* group.

In the second phase, participants evaluated the similarity between a virtual texture and its corresponding real texture. The order of the six texture pairs was randomized for each participant. The experiment was repeated three times, yielding to a total number of eighteen pairs presented to the users. Participants were instructed to explore each texture, either real or virtual, from side to side maintaining a regular scanning speed, not too fast and not too slow, within the range of 3 to 7 cm/s. In addition, they were told to apply a regular normal pressure, not too heavy and not too lite, within the range of 60 to 110 g. When interacting with the 3M panel, the users were asked to wear an anti-static wrist band to ensure that their body is electrically grounded. The bracelet had a 1 MΩ current limiting resistor to prevent excessive current passing through the user's body. A computer program, reads user's pressure and finger position every 25 ms (40 Hz) and updates the output actuation signal every 100 ms (10 Hz). A DAQ (PCI-6229, NI, USA) was used to read analog input (load-cell) and generate analog output (actuation signal) with a sampling rate of 1 kHz. For velocity estimation we employed a differentiation method based on the first order adaptive windowing (FOAW) method to obtain a reliable and stable velocity estimates from position data.

6.5.2 Results

During the experiment we collected 600 similarity scores (20 participants × (12+18) pairs). We first computed the average similarity score between different real objects (Fig. 6.10(a)) and that of identical real objects (Fig. 6.10(b)). For each pair, the average is taken between 20 scores (20 participants × 1 trial). The mean and standard deviation values of the overall similarity scores are also listed in Table 6.1. Among the different pairs, `scr-can` received the highest score (44%) and `can-chr` the lowest (9%). The total average was 30%. Among the identical pairs, chair fabric received the highest score (96%) and painting canvas the lowest (78%). The total average was 89%.

We also computed the average similarity score between a real material and its virtual counterpart (Fig. 6.11). For each of six material, the average is taken between 60 scores (20 participants × 3 trials). In the given plot, we also included lower and upper bounds (yellow bars). lower(3) indicates the average minimum among only highly-different group (3 pairs) while lower(6) indicates the average minimum among both highly-different group and moderately-different group (6



(a) Different real objects (b) Identical real objects

Figure 6.10: Average similarity scores comparing two real textures. Left: two different real materials. The red bars are for highly-different group and the blue bars for moderately-different group. Right: two similar real materials. Error bars show standard errors.

Table 6.1: The mean and standard deviation values of the similarity score between real textures.

statistics	dot-flt	pla-scr	dot-pla	can-chr	chr-flt	scr-can	total
mean	23	37	44	9	22	44	30
std	25.93	26.82	27.27	8.13	19.55	24.39	26.03
statistics	dot-dot	chr-chr	flt-flt	can-can	pla-pla	scr-scr	total
mean	89	96	90	78	94	85	89
std	11.31	6.41	14.56	18.09	8.37	16.58	14.23

pairs). The mean and standard deviation values are given in Table 6.2 and 6.3 for the real-vs-virtual scores and for the real-vs-real bounds, respectively. Plastic sheet received the highest score (70%) and chair fabric the lowest (43%). The total average was 60%.

Furthermore, to examine the statistical differences among the materials, we performed a non-parametric Kruskal-Wallis test since our similarity scores did not pass the Shapiro-Wilk normality test. The results showed that the material had a statistically significant effect on the overall similarity score ($\chi^2(5) = 19.98$, $p = 0.0013$).

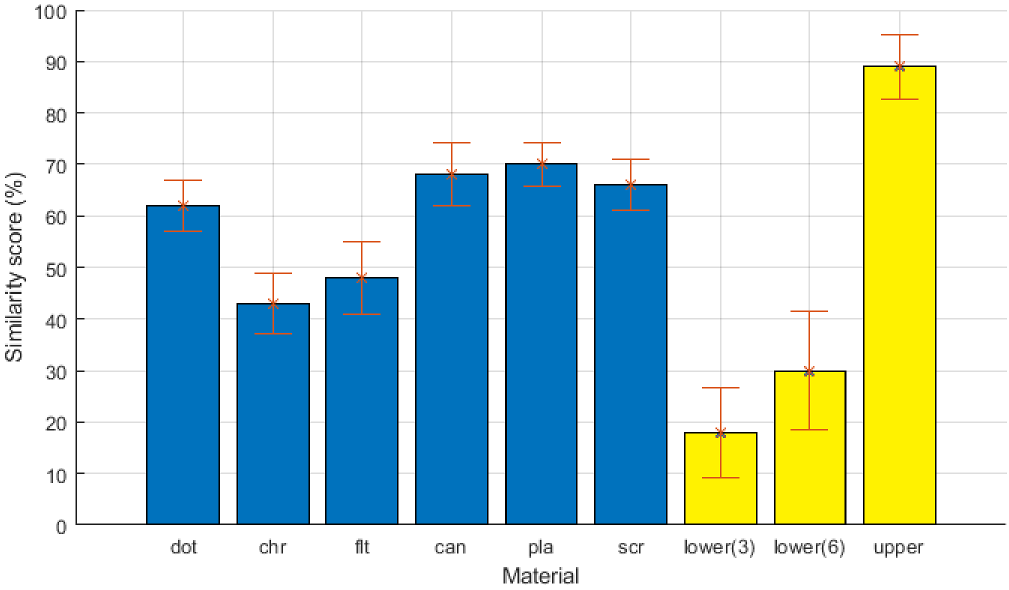


Figure 6.11: Average similarity scores comparing real and virtual textures. Error bars indicate standard errors. The yellow bars show lower and upper bounds, with lower(3) including only highly-different group (3 pairs) and lower(6) both highly-different and moderately-different groups (6 pairs).

Moreover, we calculated the average minimum score and the average maximum score among three trials for each material (Fig. 6.12). The mean and standard deviation values are given in Table 6.4. As it can be seen, there are

Table 6.2: The mean and standard deviation values of the similarity score between real and virtual textures. The values inside parenthesis are the ratios of the above values to the upper bound (89%).

statistics	dot	chr	flt	can	pla	scr	total
mean	62	43	48	68	70	66	60
	(0.70)	(0.48)	(0.54)	(0.76)	(0.79)	(0.74)	(0.67)
std	16.08	22.91	25.17	19.77	13.91	15.99	21.79

Table 6.3: The mean and standard deviation values of the lower and upper bounds.

statistics	lower(3)	lower(6)	upper
mean	18	30	89
std	20.07	26.03	14.23

occasions that virtual textures received scores as high as 95%. This means at those instances the user felt a satisfactory high similarity between the re-created virtual texture and the corresponding real one. On the contrary, there are some cases on which a similarity score as low as 10% is achieved, meaning that at those particular moments, the user felt very poor similarity between two textures.

We have also conducted a preliminary experiment to compare the performance of our proposed texture modeling and rendering algorithm with that of a basic record-and-playback method. In the record-and-playback method, the collected forces from the surface of each material under the given experimental condition (normal pressure and scanning velocity), are amplified and directly applied to the 3M panel. Same as before, we used three masses and five scanning

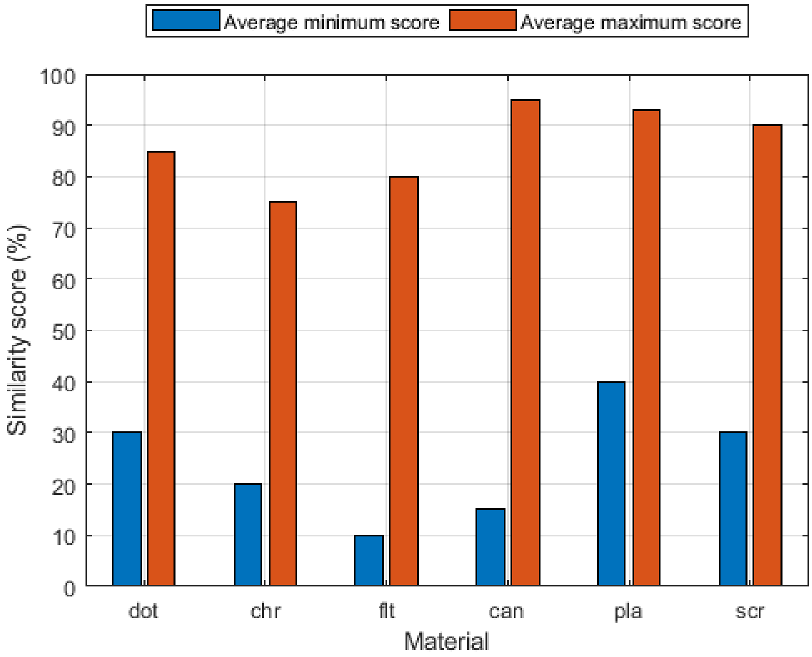


Figure 6.12: Average minimum and maximum scores for each material.

Table 6.4: The mean and standard deviation values of minimum and maximum scores between real and virtual textures.

statistics	dot	chr	flt	can	pla	scr	total
mean min	30	10	10	15	40	30	23
std min	0.00	0.00	0.00	7.07	0.00	0.00	12.15
mean max	85	75	80	95	93	90	86
std max	7.07	7.07	14.14	7.07	3.54	0.00	9.32

velocities during data collection using our tribometer. Then, the same improved two-step interpolation method is applied for the experimental conditions under which no force data is collected.

We hired 12 participants and asked them to rank the similarity between a real texture sample and its virtual counterpart rendered using record-and-playback method (Fig. 6.13). The mean and standard deviation values of the similarity score between real and virtual textures are given in Table 6.5. As it can be seen, there is a considerable difference between the average similarity scores of two methods with 60% vs. 39% for nn-based and rp-based, respectively. On average, the virtual textures generated using our proposed method are 21% more similar to the real ones than those of generated by record-and-playback method.

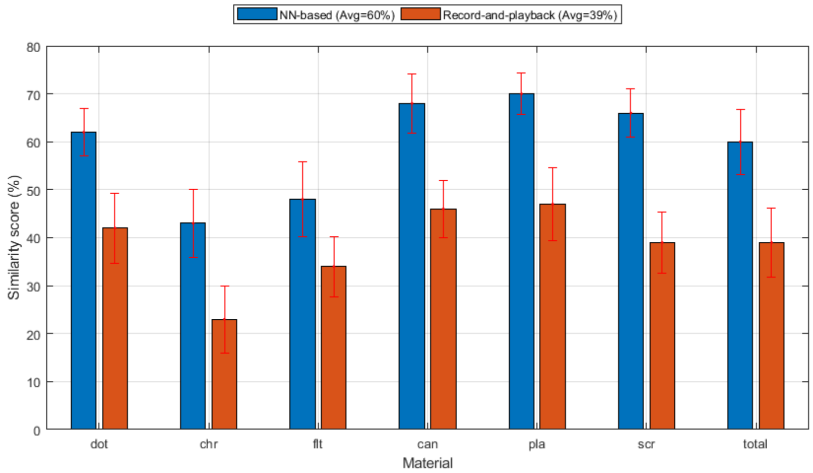


Figure 6.13: Comparison between neural network-based (nn) and record-and-playback (rp) methods. Error bars indicate standard errors.

Table 6.5: The mean and standard deviation values of the similarity score between real and virtual textures comparing nn-based and rp-based methods.

nn-based	dot	chr	flt	can	pla	scr	total
mean	62	43	48	68	70	66	60
std	16.08	22.91	25.17	19.77	13.91	15.99	21.79
rp-based	**dot**	**chr**	**flt**	**can**	**pla**	**scr**	**total**
mean	42	23	34	46	47	39	39
std	22.24	21.43	19.30	18.45	23.13	19.68	22.15

6.5.3 Discussion

Several observations can be made from the results of user study. As explained earlier, the six pairs of different real materials are categorized into two groups: highly-different group including `dot-flt`, `can-chr`, and `chr-flt` and moderately-different group consisting `pla-scr`, `dot-pla`, and `scr-can`. Accordingly, we expected to see lower scores for the former group and higher scores for the latter one. This expectation is somehow confirmed by the results. The highly-different group received an average score of 18% while the moderately-different group received 42%. `can-chr` from the first group achieved the lowest score (9%) and `scr-can` from the second group the highest score (44%). This means that participants found painting canvas and chair fabric the most different objects and scrunched paper and painting canvas the most similar objects.

Comparing real and virtual textures, Fig. 6.11 clearly shows that the similarity score of all materials are well above both lower limits. Considering only lower(3), the scores show a significant difference from this lower limit, by which we can assume that all materials were similarly rendered. This consideration is valid since it comes from the highly-different group, indicating the extent to which two very different objects are ranked to be alike. Moreover, we can also see that the scores of all materials are fairly close to the upper limit, supporting the aforementioned assertion.

Another observation can be made regarding the materials with lowest scores, i.e., chair fabric and felt fabric with 43% and 48%, respectively. Among the six real objects, these two are the ones made of a soft material. Chair fabric feels smoother than felt fabric and has some equally spaced woven patterns on its surface while felt fabric does not have any distinguishable pattern. The remaining four samples generally feel rougher than these two. It seems the users found the virtually re-created textures of these two soft materials less similar to their real counterparts. Apparently, our system performed a poor job producing the degree of softness required for better matching. Not being able to create a soft sensation was somehow expected. At the initial stage of this project, when we were testing various basic actuation signals on the 3M panel, we noticed that, in general, square waves create rougher sensation than sine waves with the same frequency. It seems the abrupt changes in the square waves are directly transformed to friction forces and perceived coarser by the moving finger. We have not observed the same behavior with smooth-changing sine waves. The estimated PRBS-like actuation signals behave like square waves which appear to have limited capacity creating softness sensation. From the opposite standpoint, these two-level signals are good at producing sharp and crisp sensations. So it is no surprise to see the highest performance is achieved for the plastic sheet (70%). The equally spaced diagonal ridges on the surface of plastic sheet can be seen as visual representation of a square wave. In addition, its rigid surface feels very similar to the glassy surface of 3M panel. It seems textures made of solid materials with rigid patterns have more chance to be realistically re-created using our system.

Lastly, comparing the neural network-based method with record-and-playback, we achieved a higher degree of similarity, 21% more given 60% vs. 39%, using our method. This is relatively 30% more than what what we could accomplish with the record-and-playback. On the one hand, this reveals that the trained neural networks could generate more appropriate actuation signals and hence more realistic virtual textures. On the other hand, this shows that when the rendering device, here the 3M panel, has limited capacity, even a simple record-and-playback method could result in some mediocre output. Nevertheless, our method notably performed better in capturing and re-creating surface fine details, emphasizing the need for incorporating the dynamic behavior of the rendering device in the modeling and rendering algorithm.

6.6 Conclusion

We proposed an effective solution to fix the problem of basic interpolation scheme introduced in the previous chapter. The main issue occurred in estimating an actuation signal from the two adjacent signals obtained under two different scanning velocities. When the interpolated signal is applied to the 3M panel, in the resulting forces we could see both components of the lower and higher velocities at two distinct frequencies instead of a single component at a frequency in between. This problem did not occur when interpolating signals under the same scanning velocity but different masses. We improved the interpolation scheme by proposing an additional process before taking weighted average. In the frequency domain, if we shift the lower component to the right to an intended frequency and the higher component to the left to the same intended frequency, then we end up with a single component at the desired frequency. Shifting in frequency domain can be done by re-sampling in the time domain, either down-sampling or up-sampling.

We proposed a two-step interpolation scheme to estimate an actuation signal based on the adjacent nodes in the pressure-velocity grid. We first take weighted average between the two signals with the same velocity but different masses. Then, we re-sample the estimated lower signal to shift its frequency component to the right and the estimated higher signal to the left. Finally, taking weighted average between them yields to an actuation signal maintaining the right frequency and power density. The key point here is to know the lower and higher frequencies and also the intended frequency. We showed a linear relation between the frequency of the main component and the scanning velocity for the examined real texture samples. This important finding, eliminates the need to know the frequencies and instead they can be replaced by the lower and higher velocities and the user applied velocity.

We put the improved interpolation scheme to the test by conducting a human user study with 20 participants. They were asked to rate the similarity between real and virtual textures on a scale of 0 to 100, with 0 being completely different and 100 being perfectly matching. As a whole, considering all six materials, an average score of 60% was achieved. Among the materials, plastic sheet received a score of 69% being the highest rated texture and chair fabric received 54% being the lowest rated texture. These results reveals the potentials of the proposed modeling and rendering method in re-creating virtual textures similar to the real ones using an electrovibration display.

We still believe there is room for improvement. For example, in this study, we focused on PRBS (pseudo-random binary sequence) to train neural networks and use them to generate a PRBS-like actuation signal for a given material under known experimental conditions. As a PRBS has only two levels, this may limit the extent to fully identify the dynamic behavior of the electrovibration display. A PRMS (pseudo-random multilevel sequence) might increase the quality of the modeling and rendering because we are no longer limited to have only two levels. This may create a finer model with higher resolution yielding to more realistic virtual textures. In addition, while the improved interpolation scheme works fine estimating signals for untested conditions, we might want to investigate a possibility for model-based interpolation. In this scenario, the parameters of a neural network is adjusted (or interpolated) based on the adjacent networks. At this moment, this is just an initial idea which we intend to address in close future.

Chapter VII.
Conclusions

In this work, we presented algorithms to render surface curvature and fine texture using a variable friction touchscreen. The motivation behind this study comes from the ever-growing interest in using touchscreen devices such as mobile phones and tablets. While technology advances improve the quality of such devices, but their functionality is still limited to simple user interactions. At best, they can generate some simple monotonic vibrations in response to user actions. A natural interaction with such touchscreen devices requires better and more intuitive (haptic) feedback, tailored based on the application. For example, an online clothing store can significantly benefit if it can provide a way to the users to touch and feel the fabric using their phones. This may look somehow futuristic but with the recent technology advances it is becoming real. Variable friction displays enable a new tactile feedback on touchscreens. Currently there are two technologies based two different phenomena: 1) based on ultrasonic vibration, and 2) based on electrovibration. They both alter the surface friction in response to user's touch. Using the former one, the surface friction decreases while with the latter one it increases. As touchscreen displays are made of a glass material with minimum friction, increasing its friction makes more sense than decreasing it. In addition, requiring less complicated hardware, makes the electrovibration technology more attractive and suitable for daily consumer electronics.

This effort is one of the first to address the possibility of realistic rendering surface features on an electrovibration display. We particularly focused on rendering surface 3D curvature and fine texture. We proposed efficient algorithms tackling each problem separately. In the first part, we introduced a gradient-based method to render 3D objects. It includes a generalized real-time algorithm to estimate surface gradient from the surface of any 3D mesh. In addition, a separate edge detection method is included to emphasis sharp edges while scanning the surface of a mesh. Conducting a human user study, we showed that in the presence of haptic feedback generated using our algorithm, the users can better recognize 3D bumps and holes when the visual feedback is limited and puzzling.

In the second part, we proposed a neural network based data-driven texture modeling and rendering method. We first created an inverse neural network dynamic model for the electrovibratin display to estimate an actuation signal from the forces collected from the surface of real texture samples. For the force measurement, we developed a linear motorized tribometer enabling adjusting applied normal pressure and scanning velocity. We trained neural networks to learn from the forces resulted from applying a full-band PRBS (pseudo-random binary signal) to the electrovibration display and generate similar actuation signals. While the networks are trained under known normal pressure and scanning velocity, for the untested conditions, we proposed a two-part interpolation scheme to produce actuation signal from the neighborhood conditions. The first part generates an estimate by taking weighted average between the signals with the same scanning velocity but different masses. The second part, performs a re-sampling process, either down-sampling or up-sampling, on the newly estimated signals to produce a final signal according to the user applied normal pressure and scanning velocity. We conducted a user study to put the proposed algorithm to test. We asked users to rate the similarity between a real texture and its virtual counterpart. The experimental setup included a load-cell to measure user applied pressure and an IR-frame to track his/her finger position and eventually calculate user's scanning velocity. Testing six different real texture samples, we achieved an average similarity score of 60%. This revealed the potentials of the proposed texture modeling and rendering algorithm accompanied by a linear interpolation scheme in creating virtual textures similar to the real ones.

References

[1] Margaret Minsky, Ouh-young Ming, Oliver Steele, Frederick P Brooks Jr, and Max Behensky. Feeling and seeing: issues in force display. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 235–241. ACM, 1990.

[2] Gabriel Robles-De-La-Torre and Vincent Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412(6845):445–448, 2001.

[3] Y. Yoo, I. Hwang, and S. Choi. Consonance of vibrotactile chords. *IEEE Trans. Haptics*, 7(1):3–13, Jan 2014.

[4] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. Teslatouch: electrovibration for touch surfaces. In *Proc. of ACM UIST*, pages 283–292, 2010.

[5] Edward Mallinckrodt, A. L. Hughes, and William Sleator Jr. Perception by the skin of electrically induced vibrations. *Science*, 1953.

[6] S Grimnes. Electrovibration, cutaneous sensation of microampere current. *Acta Physiologica*, 118(1):19–25, 1983.

[7] Robert M. Strong and Donald E. Troxel. An electrotactile display. *IEEE Trans. Man-Machine Systems*, 11(1):72–79, 1970.

[8] David J Beebe, CM Hymel, KA Kaczmarek, and ME Tyler. A polyimide-on-silicon electrostatic fingertip tactile display. In *Proc. of IEEE EMBS*, volume 2, pages 1545–1546, 1995.

[9] Hui Tang and David J Beebe. A microfabricated electrostatic haptic display for persons with visual impairments. *IEEE Trans. Rehabil. Eng.*, 6(3):241–248, 1998.

[10] Akio Yamamoto, Toshiki Ishii, and T Higuchi. Electrostatic tactile display for presenting surface roughness sensation. In *Industrial Technology, 2003 IEEE International Conference on*, volume 2, pages 680–684. IEEE, 2003.

[11] Taku Nakamura and Akio Yamamoto. Multi-finger electrostatic passive haptic feedback on a visual display. In *Proc. of IEEE WHC*, pages 37–42, 2013.

[12] 3m microutouch system (surface capacitive touch technology).

[13] David J. Meyer, Michael Peshkin, and J. Edward Colgate. Fingertip friction modulation due to electrostatic attraction. In *Proc. of IEEE WHC*, pages 43–48, 2013.

[14] Seung-Chan Kim, Ali Israr, and Ivan Poupyrev. Tactile rendering of 3d features on touch surfaces. In *Proc. of ACM UIST*, pages 531–538, 2013.

[15] Gholamreza Ilkhani, Mohammad Aziziaghdam, and Evren Samur. Data-driven texture rendering with electrostatic attraction. In *Haptics: Neuroscience, Devices, Modeling, and Applications*, pages 496–504. Springer, 2014.

[16] Heewon Kim, Jeonggoo Kang, Ki-Duk Kim, Kyoung-Moon Lim, and Jeha Ryu. Method for providing electrovibration with uniform intensity. *IEEE Trans. Haptics*, 8(4):492–496, 2015.

[17] Yang Zhang and Chris Harrison. Quantifying the targeting performance benefit of electrostatic haptic feedback on touchscreens. In *Proc. of ACM ITS*, pages 43–46. ACM, 2015.

[18] Qinglong Wang, Xiangshi Ren, Sayan Sarcar, and Xiaoying Sun. Ev-pen: Leveraging electrovibration haptic feedback in pen interaction. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, pages 57–66. ACM, 2016.

[19] J. Kang, H. Kim, S. Choi, K. D. Kim, and J. Ryu. Investigation on low voltage operation of electrovibration display. *IEEE Trans. Haptics*, 10(3):371–381, July 2017.

[20] Z Radivojevic, P Beecher, C Bower, Samiul Haque, Piers Andrew, Tawfique Hasan, Francesco Bonaccorso, Andrea C Ferrari, and B Henson. Electrotactile touch surface by using transparent graphene. In *Proceedings of the 2012 Virtual Reality International Conference*, page 16. ACM, 2012.

[21] Jukka Linjama and Ville Mäkinen. E-sense screen: Novel haptic display with capacitive electrosensory interface. In *Proc. of HAID*, 2009.

[22] Reza Haghighi Osgouei, Jin Ryong Kim, and Seungmoon Choi. Improving 3d shape recognition with electrostatic friction display. *IEEE Trans. Haptics*, 2017.

[23] Roberta L Klatzky, Sara Adkins, Prachi Bodas, Reza Haghighi Osgouei, Seungmoon Choi, and Hong Z Tan. Perceiving texture gradients on an electrostatic friction display. In *Proc. of IEEE WHC*, pages 154–158, 2017.

[24] Jin Ryong Kim, Reza Haghighi Osgouei, and Seungmoon Choi. Effects of visual and haptic latency on touchscreen interaction: A case study using painting task. In *Proc. of IEEE WHC*, pages 159–164, 2017.

[25] Tanvas.

[26] Dongbum Pyo, Semin Ryu, Seung-Chan Kim, and Dong-Soo Kwon. A new surface display for 3d haptic rendering. In *Prof. of EuroHaptics*, pages 487–495. Springer, 2014.

[27] F. Giraud, M. Amberg, and B. Lemaire-Semail. Merging two tactile stimulation principles: electrovibration and squeeze film effect. In *Proc. of IEEE WHC*, pages 199–203, April 2013.

[28] Eric Vezzoli, Michel Amberg, Frédéric Giraud, and Betty Lemaire-Semail. Electrovibration modeling analysis. In *Haptics: Neuroscience, Devices, Modeling, and Applications*, pages 369–376. Springer, 2014.

[29] Hiroshi Haga, Kazuhide Yoshinaga, Jiro Yanase, Daisuke Sugimoto, Kenichi Takatori, and Hideki Asada. Electrostatic tactile display using beat phenomenon of voltage waveforms. In *SID Symposium Digest of Technical Papers*, volume 45, pages 623–626. Wiley Online Library, 2014.

[30] Kurt A Kaczmarek, Krishnakant Nammi, Abhishek K Agarwal, Mitchell E Tyler, Steven J Haase, and David J Beebe. Polarity effect in electrovibration for tactile display. *IEEE Trans. Biomed. Eng.*, 53(10):2047–2054, 2006.

[31] Yasemin Vardar, Burak Güçlü, and Cagatay Basdogan. Effect of waveform in haptic perception of electrovibration on touchscreens. In *Proc. of EuroHaptics*, pages 190–203. Springer, 2016.

[32] Dinesh Wijekoon, Marta E Cecchinato, Eve Hoggan, and Jukka Linjama. Electrostatic modulated friction as tactile feedback: intensity perception. In *Proc. of EuroHaptics*, pages 613–624. Springer, 2012.

[33] Ian E Gordon and Victoria Morison. The haptic perception of curvature. *Attention, Perception, & Psychophysics*, 31(5):446–450, 1982.

[34] Hugh B Morgenbesser and Mandayan A Srinivasan. Force shading for shape perception in haptic virtual environments. Technical report, DTIC Document, 1996.

[35] Gabriel Robles-De-La-Torre and Vincent Hayward. Virtual surfaces and haptic shape perception. In *Proceedings ASME IMECE Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, volume 69, page 2, 2000.

[36] Margaret Diane Rezvan Minsky. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display.* PhD thesis, MIT, June 1995.

[37] Juhani Siira and Dinesh K Pai. Haptic texturing-a stochastic approach. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 557–562, 1996.

[38] Allison M Okamura, Jack T Dennerlein, and Robert D Howe. Vibration feedback models for virtual environments. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 674–679. IEEE, 1998.

[39] Steven S Wall and William Seymour Harwin. Modelling of surface identifying characteristics using fourier series. 1999.

[40] Seungmoon Choi and Hong Z Tan. An analysis of perceptual instability during haptic texture rendering. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, pages 129–136. IEEE, 2002.

[41] Raphael Hover, Gábor Kósa, Gabor Szekly, and Matthias Harders. Data-driven haptic renderingfrom viscous fluids to visco-elastic solids. *IEEE Trans. Haptics*, 2(1):15–27, 2009.

[42] Sunghoon Yim, Seokhee Jeon, and Seungmoon Choi. Data-driven haptic modeling and rendering of viscoelastic and frictional responses of deformable objects. *IEEE Trans. Haptics*, 9(4):548–559, 2016.

[43] J.M. Romano, T. Yoshioka, and K.J. Kuchenbecker. Automatic filter design for synthesis of haptic textures from recorded acceleration data. In *Proc. of IEEE ICRA*, pages 1815–1821, May 2010.

[44] Dinesh K Pai and Peter Rizun. The what: A wireless haptic texture sensor. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 3–9. IEEE, 2003.

[45] Sheldon Andrews and Jochen Lang. Interactive scanning of haptic textures and surface compliance. In *Proc. of 3DIM*, pages 99–106, 2007.

[46] S Andrews and J Lang. Haptic texturing based on real-world samples. In *Proc. of IEEE HAVE*, pages 142–147, 2007.

[47] Jochen Lang and Sheldon Andrews. Measurement-based modeling of contact forces and textures for haptic rendering. *IEEE Trans. Vis. Comput. Graphics*, 17(3):380–391, 2011.

[48] Vijaya L Guruswamy, Jochen Lang, and Won-Sook Lee. Modelling of haptic vibration textures with infinite-impulse-response filters. In *Haptic Audio visual Environments and Games, 2009. HAVE 2009. IEEE International Workshop on*, pages 105–110. IEEE, 2009.

[49] Vijaya Lakshmi Guruswamy, Jochen Lang, and Won-Sook Lee. IIR filter models of haptic vibration textures. *IEEE Trans. Instrum. Meas.*, 60(1):93–103, 2011.

[50] Joseph Romano and Katherine Kuchenbecker. Creating realistic virtual textures from contact acceleration data. *IEEE Trans. Haptics*, 5(2):109–119, 2012.

[51] H. Culbertson, J.M. Romano, P. Castillo, M. Mintz, and K.J. Kuchenbecker. Refined methods for creating realistic haptic virtual textures from tool-mediated contact acceleration data. In *Proc. of IEEE HAPTICS*, pages 385–391, March 2012.

[52] H. Culbertson, J. Unwin, B.E. Goodman, and K.J. Kuchenbecker. Generating haptic texture models from unconstrained tool-surface interactions. In *Proc. of IEEE WHC*, pages 295–300, April 2013.

[53] H. Culbertson, J.J. Lopez Delgado, and K.J. Kuchenbecker. One hundred data-driven haptic texture models and open-source methods for rendering on 3d objects. In *Proc. of IEEE HAPTICS*, pages 319–325, Feb 2014.

[54] Heather Culbertson and Katherine J Kuchenbecker. Importance of matching physical friction, hardness, and texture in creating realistic haptic virtual surfaces. *IEEE Transactions on Haptics*, 10(1):63–74, 2017.

[55] Sunghwan Shin, Reza Haghighi Osgouei, Ki-Duk Kim, and Seungmoon Choi. Data-driven modeling of isotropic haptic textures using frequency-decomposed neural networks. In *Proc. of IEEE WHC*, pages 131–138, 2015.

[56] Arsen Abdulali and Seokhee Jeon. Data-driven rendering of anisotropic haptic textures. In *Proc. of AsiaHaptics*, pages 401–407. Springer, 2016.

[57] Gholamreza Ilkhani, Mohammad Aziziaghdam, and Evren Samur. Data-driven texture rendering on an electrostatic tactile display. *International Journal of Human–Computer Interaction*, pages 1–15, 2017.

[58] Heather Culbertson, Juan Jose Lopez Delgado, and Katherine J Kuchenbecker. The penn haptic texture toolkit for modeling, rendering, and evaluating haptic virtual textures. 2014.

[59] Michaël Wiertlewski, Daniele Leonardis, David J. Meyer, Michael A. Peshkin, and J. Edward Colgate. A high-fidelity surface-haptic device for texture rendering on bare finger. In *Haptics: Neuroscience, Devices, Modeling, and Applications*, pages 241–248. Springer, 2014.

[60] Eric Vezzoli, Thomas Sednaoui, Michel Amberg, Frédéric Giraud, and Betty Lemaire-Semail. Texture rendering strategies with a high fidelity-capacitive visual-haptic friction control device. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 251–260. Springer, 2016.

[61] David A Forsyth and Jean Ponce. Computer vision: A modern approach. *Computer Vision: A Modern Approach*, pages 88–101, 2003.

[62] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

[63] Xunnian Yang, Jianmin Zheng, and Desheng Wang. A computational approach to joint line detection on triangular meshes. *Engineering with Computers*, 30(4):583–597, 2014.

[64] Andreas Hubeli, Kuno Meyer, Markus H Gross, Markus H Gross, and Markus H Gross. *Mesh edge detection.* Swiss Federal Institute of Tech-

nology, Department of Computer Science, Institute of Scientific Computing, Computer Graphics Group, 2000.

[65] Unity - game engine.

[66] Lennart Ljung. *System identification: theory for the user.* Prentice Hall, 1998.

[67] Fouad Giri and Er-Wei Bai. *Block-oriented nonlinear system identification*, volume 1. Springer, 2010.

[68] Robert Haber and László Keviczky. *Nonlinear system identification-Input-output modeling approach.* Kluwer Academic Publishers, 1999.

[69] Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models.* Springer Science & Business Media, 2013.

[70] Er-Wei Bai. Identification of linear systems with hard input nonlinearities of known structure. *Automatica*, 38(5):853–860, 2002.

[71] IJ Leontaritis and Stephen A Billings. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *Int. J. Control*, 41(2):303–328, 1985.

[72] Ho Pham Huy Anh and Kyoung Kwan Ahn. Hybrid control of a pneumatic artificial muscle (pam) robot arm using an inverse narx fuzzy model. *Eng. Appl. Artif. Intell.*, 24(4):697–716, 2011.

[73] HK Sahoo, PK Dash, and NP Rath. Narx model based nonlinear dynamic system identification using low complexity neural networks and robust h-inf filter. *Appl. Soft Comput.*, 13(7):3324–3334, 2013.

[74] 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 & motor research*, 24(1-2):53–70, 2007.

[75] 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, & Psychophysics*, 65(4):613–631, 2003.

[76] E&T Tech infrared touch screen.

[77] Cas scale. `http://www.casscale.co.kr`.

[78] Seokhee Jeon, Seungmoon Choi, and Matthias Harders. Rendering virtual tumors in real tissue mock-ups using haptic augmented reality. *IEEE Trans. Haptics*, 5(1):77–84, 2012.

Acknowledgements

I greatly appreciate all the help and support I received from my adviser, Prof. Seungmoon Choi, during the last seven years in Haptics and Virtual Reality Laboratory. As a foreigner living in a culturally different but peaceful country, I had ups and downs, but he patiently assisted me to go through the difficulties and bring the best out of me. I am also thankful to my colleagues and lab-mates, they made my living and academic life easier by offering their unlimited kindness. I have to thank my family too, my beloved wife and son; they kindly believed in me and tolerated my busy schedule. Finally, I would like to thank my precious parents who unconditionally supported me and gave me their love and affection no matter what. I wish my father was alive to see the achievements of his son and to be proud of him.

Curriculum Vitae

Name : Reza Haghighi Osgouei

Education

2013. 9. – 2017. 12.	Department of Computer Science and Engineering, Pohang University of Science and Technology (Ph.D.)
2010. 3. – 2012. 2.	Department of Computer Science and Engineering, Pohang University of Science and Technology (M.Sc.)
2000. 9. – 2002. 6.	Department of Electrical Engineering, Tehran Polytechnic (M.Sc.)
1994. 9. – 1999. 6.	Department Electrical Engineering, Sahand University of Technology (B.Sc.)

Experience

2012. 3. – 2013. 8.	Researcher at HVRLab., POSTECH, Rep. of Korea
2006. 2. – 2009. 8.	Researcher at JMERC, Iran

2004. 9. – 2005. 8. Research assistant at IAU, Iran

2002. 9. – 2004. 8. Electrical engineer at Rasa Co., Iran

Affiliation

1. Haptics and Virtual Reality Lab., Department of Computer Science and Engineering, Pohang University of Science and Technology