Creating Realistic Virtual Textures from Contact Acceleration Data

Article *in* IEEE Transactions on Haptics · April 2012

DOI: 10.1109/TOH.2011.38 · Source: IEEE Xplore

Joseph Michael Romano
University of Pennsylvania

Creating Realistic Virtual Textures from Contact Acceleration Data

Joseph M. Romano, *Student Member, IEEE,* and Katherine J. Kuchenbecker, *Member, IEEE*

Abstract—Modern haptic interfaces are adept at conveying the large-scale shape of virtual objects, but they often provide unrealistic or no feedback when it comes to the microscopic details of surface texture. Direct texture rendering challenges the state of the art in haptics because it requires a finely detailed model of the surface's properties, real-time dynamic simulation of complex interactions, and high-bandwidth haptic output to enable the user to feel the resulting contacts. This paper presents a new, fully-realized solution for creating realistic virtual textures. Our system employs a sensorized handheld tool to capture the feel of a given texture, recording three-dimensional tool acceleration, tool position, and contact force over time. We reduce the 3D acceleration signals to a perceptually equivalent one-dimensional signal, and then we use linear predictive coding to distill this raw haptic information into a database of frequency-domain texture models. Finally, we render these texture models in real time on a Wacom tablet using a stylus augmented with small voice coil actuators. The resulting virtual textures provide a compelling simulation of contact with the real surfaces, which we verify through a human subject study.

Index Terms—haptic texture rendering, virtual reality, data-driven modeling, high-frequency vibrations

1 Introduction

Texture is a fundamental surface property that shapes our haptic perception in almost every real-world interaction. The feel of a pencil writing on paper, fingers stroking a piece of fabric, or a saw cutting through wood are all examples of interactions with microscopic surface irregularities [1]. The unique vibrations generated by each of these interactions give us vital information that can be used to understand objects and complete everyday tasks. Therefore, these cues also have the potential to contribute to the realism and immersiveness of virtual environments.

Modern haptic hardware and algorithms are proficient at portraying the global shape and stiffness of objects using low-frequency forces. However, they are seldom capable of accurately reproducing the high-frequency accelerations that occur during real-world contacts [2], [3]. Haptic hardware typically does not have the bandwidth to display these vibration signals, e.g., [4], and most software algorithms make no attempt to generate them. While many methods have been proposed for representing textures, no commonly accepted solution exists [5].

This article presents methods for recording, modeling, and recreating tool-mediated texture contact vibrations. A sensorized handheld tool is used to capture contact acceleration signals and tool forces and motions during interaction with a set of sample textures (Fig. 1). We distill the haptic recordings into a database of frequency-domain texture models. Finally, we render this information using a custom tablet-based haptic device.

• *J. M. Romano and K. J. Kuchenbecker are with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104.*
E-mail: {jrom, kuchenbe}@seas.upenn.edu

Fig. 1. The eight materials that were haptically modeled with the methods presented in this article. Each surface has unique microscopic details that induce distinctive vibrations when contacted with a probe.

1.1 A Psychophysically Motivated Approach

Texture information is so important to how people interact with objects that researchers have labeled it as one of the primary means of human object exploration [6]. When texture exploration is done via an intermediary tool, rather than direct skin-to-texture contact, the vibrations of the tool play an important role in the sensory experience [7]. Researchers believe that humans use total spectral power to distinguish texture vibration signals from one another [8]. For tool-mediated contact, Yoshioka et al. [9] point out that these vibrations change mainly as a function of the **normal force** between the tool and the surface and the **scanning speed** of the tool across the surface.

The methods described in this paper focus on reproducing the frequency-specific power of real vibration signals in a virtual environment. The spectrum of the created signal needs to vary with the instantaneous normal force and scanning speed of the virtual interaction. We hypothesize that reproducing these signals accurately during natural human motions will cause virtual textures to feel like their real counterparts.

1.2 Interactive Pen Display

To focus on two-dimensional haptic texture information, this work uses a custom haptic device based on a tablet computer. We chose an interactive pen display for its high stiffness in the normal direction, collocated graphics and haptics, and integrated force and position sensing. We augmented the pen with voice coil actuators to enable high-frequency vibration output.

Our goals were to capture and recreate real tool-texture vibrations and study how well precisely controlled vibrations can simulate texture contact in a virtual environment. Past approaches in pen-based haptic displays include the Ubi-Pen [10], a pen that housed a miniature tactile pin-array and small eccentric-mass vibration motor, and the haptic pen [11], a solenoid-actuated device. Our design uses voice coil actuators to achieve finer control over vibration frequency and amplitude than these alternative actuation schemes. We have taken inspiration from the experimental designs of both Kontarinis and Howe [13] and Wall and Harwin [12] to develop a high-fidelity vibration output system that can realistically render our tool-mediated texture models.

2 Background

Generating a haptic representation of real surface texture is a difficult task because of three major challenges. First, real texture-based haptic information arises as a result of microscopic tool-surface interactions, but it is challenging to record the small-scale topography and interaction properties of a tool and a surface. Second, even if these critical physical properties were measured, it is not likely that the simplistic dynamic models in use today could capture all the subtleties of the micro-surface interaction. We believe that many of the assumptions used in previous texture-rendering approaches, such as constant contact and non-deforming surface profiles, are fundamentally invalid. Third, even if such a model existed, the computational complexity of such a simulation would be too costly for real-time interaction [5].

Many different approaches have been suggested to overcome the problem of unknown three-dimensional micro-surface geometry. Basdogan et al. [14] developed a technique for inferring surface shape based on gray-scale surface images. Another popular approach is to use parametric functions, such as sinusoidal waveforms [15], to create a non-uniform haptic surface shape. In another variation, Costa and Cutkosky [16] used a fractal approach to simulate the surface of natural objects, such as rocks. Crossan et al. described a method in which granular sections of surface geometry could be combined to form unique texture profiles [17]. Both Andrews and Lang [18] and Wall and Harwin [19] used recorded topographical surface data to create Fourier series models of texture patterns. This position disturbance pattern was later replayed to the user to simulate a virtual interaction. While neither reports a detailed user study, Wall and Harwin do note that many users commented that “the surface did not resemble its real life counterpart.”

Other methods for avoiding the computational complexity of tool-surface interaction often ignore surface shape altogether and focus instead on the sensation the user should experience. Minksy et al. [20], [21] showed that surface roughness could be simulated by applying tangential repelling and attracting forces as the haptic probe slid across a flat surface. Additional work has been done to automatically generate Gaussian-distributed force fields for this same rendering approach [22], [23]. Additionally, Kyung and Lee showed that their Ubi-Pen was capable of displaying texture sensations based upon tactile gradients designed from basic shape-primitives [10]. However, little research has been done to relate these textures to real-world surfaces. They are often hand-created by the programmer as an interpretation of what a real surface might feel like.

Simultaneously, research on data-driven haptic modeling of other surface properties has shown promise in overcoming some of the difficulties in traditional physics-based haptic display. Several researchers have used data captured from real interactions to create virtual haptic interactions that feel very similar to their real counterparts. For example, real-world contact data has been used directly for playback of cutting [24], tapping [3], and clicking [11], and it has been used indirectly to set the parameters of models for a toggle switch [25] and the texture of soil [26]. 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 vaguely remembered haptic sensation. Instead, the goal of the modeling process is to capture the output response of the system (e.g., force, acceleration) given some set of user inputs (e.g., position, velocity, 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.

In the area of data-driven texture modeling, some of our previous research [27], along with that of [28], [29], has shown that the high-frequency vibrations created during interaction with real textures can be modeled and used later for virtual recreations. The type of texture model varies, but they all focus on the high-frequency forces or accelerations that the user feels when contacting the surface. In parallel, older research showed that high-frequency information is critical for high-fidelity haptic feedback in teleoperation [13], where the user controls the movement of a remote robot. In our own recent work comparing various teleoperation controllers, we showed that matching high-frequency vibrations yielded the most realistic contact experience for the user [30]. This article aims to show that similar benefits can be achieved in virtual environments by creating texture vibration models from real contact data and carefully controlling the high-frequency vibration output to be appropriately correlated with the user's actions.

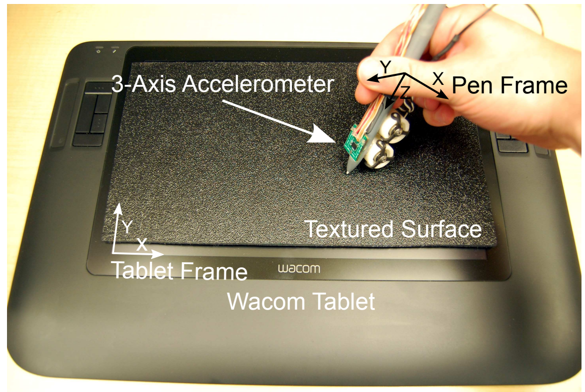

Fig. 2. Our data-collection system. The experimenter explores the textured surface with the pen, varying their normal force and scanning speed. The Wacom tablet measures the pen's normal force and position, while the three-axis accelerometer measures its vibrations.

3 Creating a Texture Model

The first step in creating a texture model is to capture haptic data from the desired interaction and distill it into a frequency-domain model. This Section describes our texture-capture hardware (Fig. 2) and software (Fig. 3), along with the capturing procedure.

3.1 Data Collection System

Our data collection system centers on a computer connected to an interactive pen display and a digital three-axis accelerometer (Fig. 2). These sensors capture the normal force, scanning speed, and high-frequency accelerations generated when the pen interacts with various texture samples laid on the screen.

3.1.1 Interactive Pen Display

We use a Cintiq 12WX interactive pen display by Wacom Co., Ltd. to determine the tool's normal force and scanning speed during data collection. This particular Wacom model was selected for several reasons, including its lightweight pen, collocated graphics, and high-resolution force and position measurement. The model of pen used to interact with the display surface is a Cintiq Classic Pen.

A modified version of the Linux Wacom Project code base [31] was used to gain access to the raw data sensed by the Wacom device. Three variables are read from the Wacom device as positive integer count values at a rate of 125 Hz: the x- and y-position of the pen tip (both in the tablet frame), and the pen's normal force. The two-dimensional position of the pen tip can be converted to real-world units by multiplying by the position resolution as defined by the manufacturer: (4.934 μm/count) and (4.861 μm/count) for the x- and y-axes respectively. The normal force sensor, located in the tip of the Wacom pen, has a resolution of (0.0013 N/count).

The velocity at which the pen tip translates over the Wacom surface is calculated via the discrete-time derivative of tip position. The pen speed is calculated by taking the magnitude of this two-dimensional velocity vector. During data collection we apply a 1 Hz first-order low-pass filter to these speed values to diminish the effects of noise and quantization in the tablet position readings.

3.1.2 Three-axis Accelerometer

An Analog Devices ADXL345 three-axis digital MEMS-based accelerometer is affixed to the pen to capture its high-frequency vibrations. This chip was selected for its ability to read accelerations along three Cartesian axes, its high bandwidth, and its configurable range of up to ± 157 m/s^2 (± 16 g). We use a DIMAX SUB-20 Multi Interface USB Adapter to connect the ADXL345 to the laptop computer.

Our custom data collection software gathers acceleration data from the ADXL345 using a C library supplied by DIMAX. Upon start up, our software configures the ADXL345 into ± 78.4 m/s^2 (± 8 g) mode and sets its sampling rate to 800 Hz. During operation, the program polls the ADXL345 at 800 Hz and records the x-y-z acceleration values. The ADXL345 can distinguish between 2^{10} values of acceleration on each axis, and it returns each reading as an integer value. We subtract calibrated zero acceleration values and convert to standard units by multiplying by the resolution of the accelerometer, 0.153125 (m/s^2)/count.

3.1.3 Texture Samples

Eight common materials were selected for our initial set of textures: vinyl, denim, canvas, wood, rough plastic, paper, brushed plastic, and cardboard (Fig. 1). Each sample was mounted with spray adhesive on a piece of 2.25-mm-thick plastic. These materials were selected because the vibrations generated felt the same regardless of the scanning direction. This isotropy was verified by recording the vibrations produced while moving the tool over each material in a variety of directions. Each of the selected materials produced a spectrum that was largely invariant to the scanning direction. Furthermore, none of the materials had features that were spatially perceptible during pen contact, such as large bumps or crevices, so they feel the same at all locations on the surface.

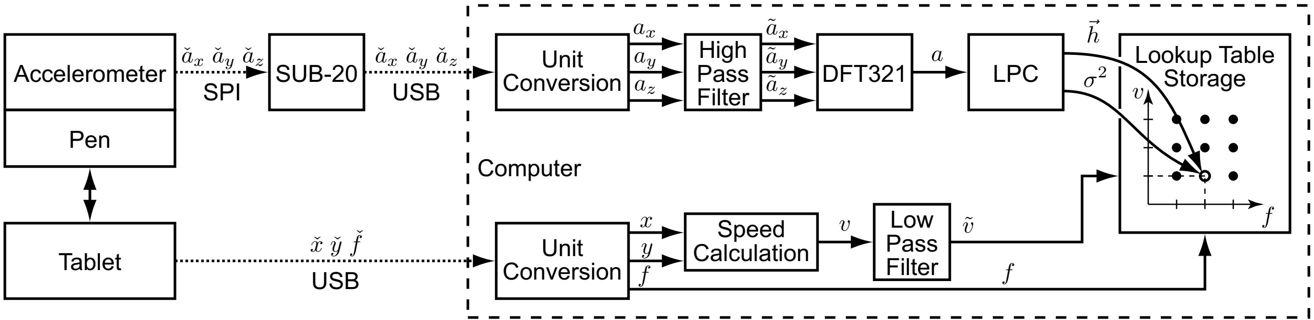

Fig. 3. A diagram of our texture modeling system. Three-axis acceleration data is read from the pen-mounted accelerometer into the computer. The acceleration information is high-pass filtered to remove the low-frequency effects of gravity and human motion. The DFT321 algorithm condenses the three-axis acceleration into a one-axis waveform. We then use linear predictive coding to develop a transfer-function model of this signal, and we store the results in a lookup table at the proper speed and force indices. Pen-tip position and normal force are read from the Wacom tablet into the computer, and the position values are used to calculate pen speed.

3.2 Data Collection Procedure

Each material was manually scanned at sixteen combinations of normal force and speed using the procedure outlined below. We scan materials manually rather than with a robot to avoid the complexity of automation and ensure the tool has an appropriate dynamic response.

First, a texture sample is placed on top of the Wacom device as seen in Fig. 2. The Wacom is capable of recording position and force data through the surface. Next, the experimenter grasps the instrumented stylus comfortably in their hand and runs our custom data collection software program. The program displays the current pen force and speed on the screen, and it gives the experimenter ten seconds to adjust their movement before acceleration recording commences. The experimenter pushes harder or softer on the surface to control the force to the desired value. To regulate speed, the experimenter moves the pen in a circular motion around the material at slower or faster rates. The experimenter attempts to hold the pen in a vertical orientation throughout the trial. After recording six seconds of data, the program exits and writes the data to a file.

For each of the eight surface materials, data was recorded at intervals of 0.3 N from 0.3 to 1.2 N, and at intervals of 50 mm/s from 50 to 200 mm/s, for a total of sixteen recordings per material. The ranges were selected to span the entire space of reasonable force and speed values a user generates when interacting with the Wacom device. The intervals were selected so that the experimenter could easily achieve each of the predefined values. If the experimenter deviated more than 0.15 N or 25 mm/s away from the target force or speed for a duration greater than 0.5 seconds, the recording trial was discarded and repeated.

3.3 Texture Response Modeling

After capturing these three-axis vibration signals, we need to store the information they contain. The rest of the methods discussed in this Section do not need to run in real time, so they have been implemented offline using Matlab. For reasons that will quickly become clear in Section 4, reducing our raw data to a frequency-domain model is very useful for creating a unique and infinitely long version of the signal in a haptic virtual environment.

3.3.1 Acceleration Data Pre-Processing

Prior to our main data manipulation steps, we perform several simple pre-processing steps to condition our acceleration data. First, we upsample all data from its original recorded 800 Hz to 5000 Hz using the `resample` function in Matlab. We upsample because we use the computer soundcard to output our actuator drive signal, and 5000 Hz is a standard minimum output rate available in many soundcards. Next, we high-pass filter our data with a cutoff frequency of 10 Hz. This step eliminates several unwanted effects including gravitational forces, sensor drift, and accelerations from the experimenter's hand motions.

3.3.2 Converting to a One-Dimensional Signal

Many researchers have noted that the human sensitivity to high-frequency vibrations is largely independent of direction [32]. Additionally, making a high-frequency vibration haptic device with one output axis is far simpler than making a device with three axes of output. These two factors have motivated us to develop a method of reducing the vibration information recorded in three Cartesian directions to a single dimension. The goal of this reduction is to create a vibration signal that has little or no perceptual difference from the original three-dimensional signal. The proper way to combine these three signals is not immediately evident, and our research group

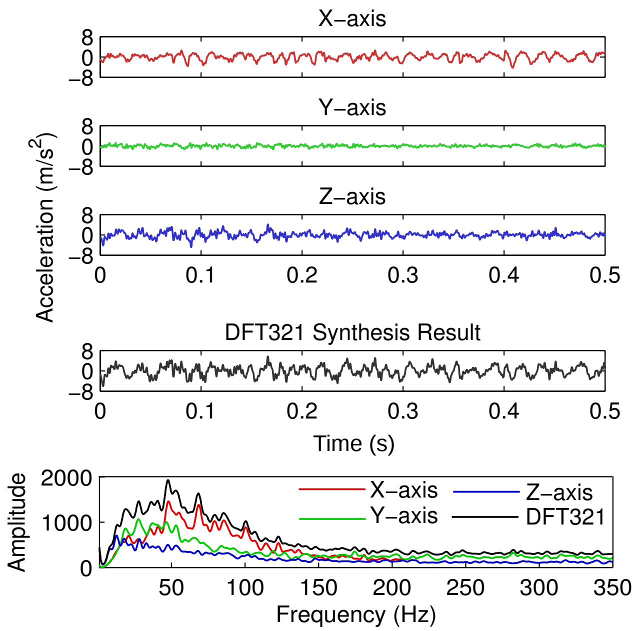

Fig. 4. Three-axis time-domain signals are reduced to one axis using DFT321. The new signal maintains important temporal features (fourth plot), and the same total spectral power (fifth plot). The depicted signals were recorded during an interaction with canvas at a speed of 50 mm/s and a normal force of 0.3 N.

has spent considerable effort exploring various approaches [33]. For offline processing, our DFT321 algorithm best captures both the spectral energy and the temporal information from all three axes, as shown for a sample recording in Fig. 4.

The DFT321 approach uses frequency-domain techniques to combine the three original signals into one signal without altering the total spectral power. First, we square, sum, and square-root the magnitude of the smoothed Discrete Fourier Transform (DFT) of each of our original three signals:

$$\left|\tilde{A}_s\left(f\right)\right| = \sqrt{\left|\tilde{A}_x\left(f\right)\right|^2 + \left|\tilde{A}_y\left(f\right)\right|^2 + \left|\tilde{A}_z\left(f\right)\right|^2} \qquad (1)$$

Here, $\left|\tilde{A}_s\left(f\right)\right|$ is the frequency-domain magnitude of our new DFT321 signal, and $\tilde{A}_x\left(f\right)$, $\tilde{A}_y\left(f\right)$, and $\tilde{A}_z\left(f\right)$ are the smoothed DFTs of each of our original three Cartesian acceleration vectors. The frequency-domain phase of our new DFT321 signal is determined by taking the inverse tangent of the sum of the imaginary parts divided by the sum of the real parts of the DFTs of the original signals, yielding an average phase $\theta(f)$:

$$\theta(f) = \tan^{-1}\left(\frac{Im\left(\tilde{A}_x\left(f\right) + \tilde{A}_y\left(f\right) + \tilde{A}_z\left(f\right)\right)}{Re\left(\tilde{A}_x\left(f\right) + \tilde{A}_y\left(f\right) + \tilde{A}_z\left(f\right)\right)}\right) \qquad (2)$$

We perform an inverse DFT on the calculated magnitude and phase to create the new time-domain signal.

3.3.3 *Generating a Linear Predictive Model*

After reducing our data to a single axis, we store it in the form of a frequency-domain transfer function. We first proposed this idea in [34], and we developed the details further in [27]. This step is summarized below, and the reader should refer to these prior works for a more detailed explanation.

Our goal in this step is to develop a discrete-time transfer function that can predict the next sample in a texture acceleration stream based on the previous n acceleration values. We approach this problem by borrowing a well-known tool from the speech processing community, Linear Predictive Coding (LPC). Let our acceleration data vector from DFT321 be called $\vec{a}(k)$, the prediction of our filter be $\vec{\hat{a}}(k)$, and the residual of these two signals $\vec{e}(k)$. $H(z)$ is defined as an IIR filter of length n of the form $H(z) = [-h_1 z^{-1} - h_2 z^{-2} ... -h_n z^{-n}]$. The transfer function from acceleration to prediction error can be written as:

$$\frac{E(z)}{A(z)} = 1 - H(z) \qquad (3)$$

We can then define the vector of filter coefficients $\vec{h} = [h_1\ h_2\ h_3\ ...\ h_n]^T$, and we can write out the residual at each time step with the following difference equation:

$$e(k) = a(k) - \hat{a}(k) = a(k) - \vec{h}^T \vec{a}(k-1) \qquad (4)$$

As described in [27], we solve for the optimal filter vector, $\vec{h}_o$, that corresponds to a minimum value of $e(k)$. We have found that a filter order of $n = 400$ generally achieves the maximum possible level of signal prediction accuracy for 5000 Hz vibration data.

Another important result of this modeling step is the residual error signal, $e(k)$, that remains after subtracting the prediction from the original signal. The power in this signal, σ^2, is calculated using the standard definition as the average squared magnitude of the individual elements in $\vec{e}(k)$.

3.3.4 *Creating Models For All Forces and Speeds*

The above process of reducing our vibration data to a single axis with DFT321 and making a transfer function model with LPC is repeated for each data recording. The resulting LPC models are then used to construct a table of model parameters for each material sampled. We parametrize the space based on the force and speed values from the recording. As shown in Fig. 3, each node in the table is composed of the LPC recursive coefficients, $\vec{h}$, and the power of the residual signal, σ^2. A Matlab script writes this data as a custom C++ lookup table class for use by the algorithms of Section 4. For rendering, it is also desirable to make table entries for the cases when force and/or speed are equal to zero. However, no useful data can be recorded under zero force or speed conditions. Thus, the zero force and speed nodes are assigned the same $\vec{h}$ as the closest node, with a σ^2 value of zero.

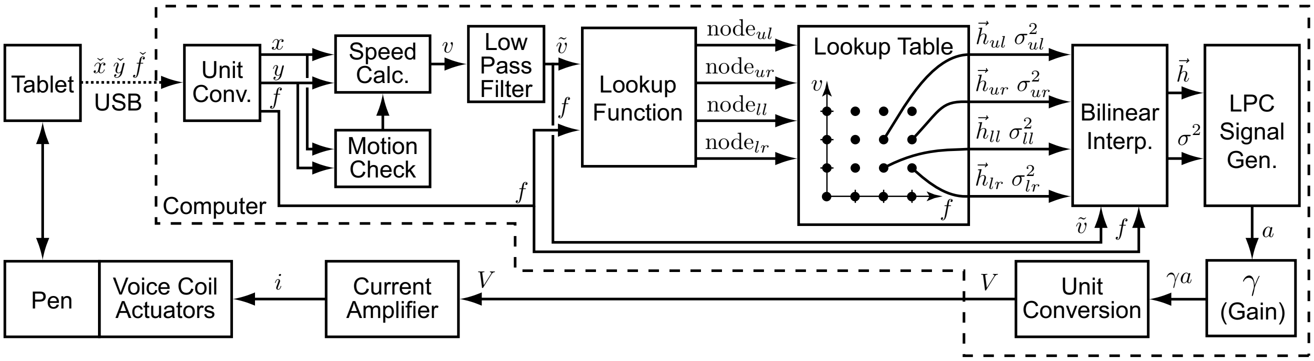

Fig. 5. A diagram of our texture rendering system. Pen-tip position and force information are read into the computer from the Wacom tablet. The position information is differentiated and low-pass filtered to obtain the pen-tip speed. The force and filtered speed information are used by our lookup table to obtain the indices of the four closest nodes in the discretized force-speed workspace. We use the models stored in these nodes to perform a bilinear interpolation and estimate the current coefficient vector and residual variance. These values are passed to our LPC signal generator, which creates an appropriate acceleration waveform. This signal is output via the computer soundcard, through a current amplifier, to drive the voice-coil actuators on the pen.

4 Rendering a Texture Model

After obtaining a data-driven model of each texture's vibration response at various speeds and forces, we use these models to create an acceleration signal for contact with virtual textures. This Section describes the process used to go from the frequency-domain model developed in Section 3 to a unique time-domain signal during a virtual texture interaction. We also describe the custom hardware system that allows the user to feel this vibration signal. Fig. 5 shows the full diagram of the texture rendering process.

4.1 Vibration Signal Generation

The first step in rendering a realistic virtual texture is to decide what sensation the user should experience at each instant in time. Several important pieces of information are required to do this: the texture type, the normal force, and the scanning speed.

Our software determines which texture the user is contacting via two-dimensional collision detection between the pen tip and textured patches on the Wacom device screen. The normal force is read directly from the Wacom device as explained in Section 3.1.1. The Wacom's raw position data is again used to calculate speed, but with a slightly different algorithm than Section 3.3.1 because the speed can no longer be assumed to be a near constant value. A first-order low-pass filter with a cutoff frequency of 20 Hz is applied to our speed value to smooth quantization effects. This higher cutoff frequency avoids introducing significant delay in the speed calculation because we want the system to respond quickly to user motion.

It is also important to note that the Wacom device has a small amount of additive noise that contaminates the position data when the pen tip is stationary. This noise generates incorrect non-zero speed values. We remove this error by high-pass filtering the x-y position values at 10 Hz to remove the DC position offset, and then low-pass filtering the signals at 100 Hz to attenuate the additive noise. If this new filtered position value is less than the empirically tuned threshold of 0.06 mm, the speed is set to 0 mm/s.

4.1.1 Lookup-Table Data Retrieval

The pen speed and force data are used in real time to retrieve data from the lookup table that was described in Section 3.3.4. The user's speed value is rounded to the two closest speeds that exist within the lookup table, one lower and one higher than the actual speed. The same procedure is done for the user's force value. These four values are used to retrieve the four closest nodes within the lookup table.

The program then uses the original values of the user's force and speed to perform a two-dimensional bilinear interpolation to determine new $\vec{h}$ and σ^2 values for the current state. The interpolated results are used to generate a unique acceleration signal, as discussed in the next Section. Note that a special case exists when the user exceeds the maximum lookup table value for force and/or speed. When this happens, we saturate the user's force or speed to the maximum value in the lookup table to avoid extrapolation.

4.1.2 LPC Signal Synthesis

The next step is to use the interpolated LPC model parameters to generate an appropriate acceleration waveform. The interpolated coefficients $\vec{h}$ are used to create a transfer function from prediction error to acceleration, which is the inverse of (3):

$$\frac{A_g(z)}{E_g(z)} = \frac{1}{1 - H(z)} \tag{5}$$

This transfer function can be used to convert white noise into an acceleration signal with the desired

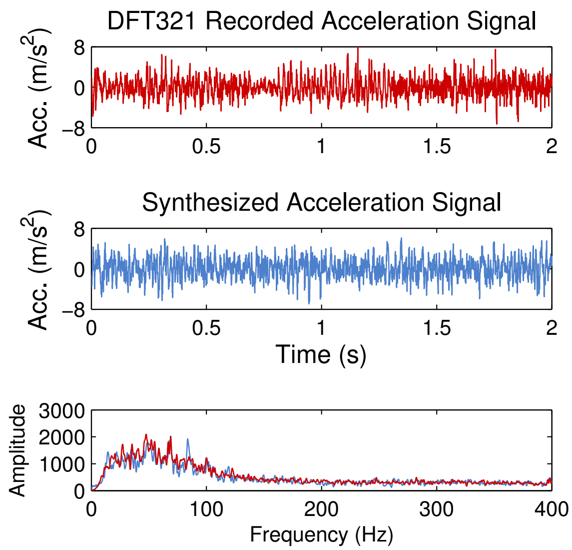

Fig. 6. A real recorded acceleration signal (top) versus a version of this signal synthesized using LPC (middle). The two signals are temporally distinct, but their highly similar spectra (bottom) make them feel almost indistinguishable.

spectrum. The difference equation for the synthesized acceleration is:

$$a_g(l) = e_g(l) + \vec{h}^T \vec{a}_g(l-1) \tag{6}$$

During texture synthesis our software generates a white noise signal, $e_g(l)$, to pass into the filter described in (5). The average signal power of the white noise excitation is of critical importance for controlling the magnitude of the resulting acceleration signal. To synthesize a texture at a specific normal force and scanning speed, we ensure that the power of the generated noise signal is equivalent to the interpolated noise power, σ^2, from the lookup table. Fig. 6 shows a sample signal generated in this way, along with the real texture vibration that was used to create the model.

4.2 Rendering Hardware

Now that we can calculate the acceleration signal we want the user to experience, we use our custom-designed hardware system to output this signal and vibrate the pen in the user's hand.

First, the magnitude of the desired acceleration signal is scaled by gain K with units of A/(m/s^2) to account for the inverse dynamics of the pen-actuator system (discussed further in Section 4.2.1), and then a conversion factor of 1 V/A is applied to account for the linear current amplification hardware. Next, we scale this output by the dimensionless parameter γ, an adjustable gain used in the experiments of Section 5. This software signal is transformed to a soundcard output value (sound output) by multiplying by a scalar conversion ratio with units of sound output/V. Next, this signal, represented as a floating point number, is sent out of the laptop soundcard using the PortAudio C library [35]. The relationship between the output count value and the soundcard output voltage was experimentally determined to be a linear ratio of 1.25 V/ sound output. This voltage is then passed into a linear current amplifier that has a gain of 1 Amp/V. The output of the linear current amplifier drives two voice coil actuators on the pen.

4.2.1 Voice Coil Actuators

The stylus is augmented with two voice coil actuators wired in series, both model NCC01-04-001-1X by H2W Technologies. Voice coils were selected as the actuation mechanism because of their direct relationship between input current and output acceleration [36]. Both voice coils are driven simultaneously in the same direction. Two voice coils are used to double the output force, a necessity for achieving the strong vibrations of surfaces such as the rough plastic used in our experiments. The actuators are mounted with the electromagnetic coil fixed and the permanent magnet moving, as shown in Fig. 7. A small weight is attached to the magnet to strengthen the system's vibration output. The total mass of the moving components is 7 grams for each actuator. A custom housing was made to mount each moving magnet and stationary electromagnetic coil to the pen. The housing contains a low-friction acetal plastic sleeve bearing that the moving magnet slides within. A rubber flexure spring is attached to the moving magnet to return it to a centered position when no drive current is applied.

Several actuator mounting positions were tested, and our final design places both actuators below the grip of the user, as seen in Figs. 2 and 7. In this configuration the actuators vibrate along the pen frame's x-axis, parallel to the tablet surface. This placement was selected because it makes the vibrations feel as if they are emanating from the tip of the pen without significantly blocking the user's view of the tablet screen. Actuator placements above the user's hand perpendicular to the z-axis of the pen tend to make vibrations feel unnatural and also create spurious moments about the user's grip. Actuator placements that align the actuators' vibration axes with the pen frame's z-axis are not capable of generating accelerations down into the rigid surface, and thus the acceleration effects are much weaker.

We experimentally identified the dynamics of our hand-pen-actuator system to better understand its high-frequency output capabilities. Two different users held the stylus in their hand while a swept sinusoid signal was played through the vibration actuators. The resulting vibrations were recorded along the x-y-z axes of the accelerometer. Five iterations of the sinusoid were repeated in sequence, each six seconds long and increasingly logarithmically from 5

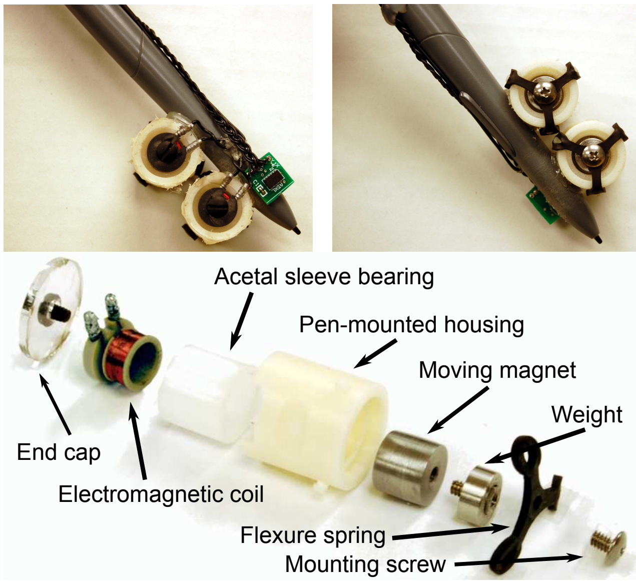

Fig. 7. Mounted (top) and exploded (bottom) views of our voice coil actuators. The end cap attaches the electromagnetic coil to the housing. Inside of the housing is a low-friction acetal sleeve bearing that the moving magnet runs within. The outside of the housing has three mounting hooks that the rubber flexure spring is attached to. A mounting screw fixes the center of the flexure spring to the moving magnet and weight.

to 300 Hz. Fig. 8 shows a plot of both the amplitude and phase response of the system for two tests with each user. The input signal is the current used to drive the actuators measured in amps (A). The output signal is the recorded acceleration vector reduced to a single axis using the DFT321 algorithm. Due to uncertainty in sound output timing, the signals were shifted to minimize time delay in each case. The system delivers a strong vibration output from approximately 30 Hz to 300 Hz, with slight variations between trials and users. As seen in Fig. 8, the frequency response of the system appears to converge to a value between 9 and 12 $(\mathrm{m/s^2})/\mathrm{A}$, indicating that a proper value for K lies between 1/9 and 1/12. We use a value of $K = 0.0972$ $\mathrm{A}/(\mathrm{m/s^2})$ for the experiments of Section 5.

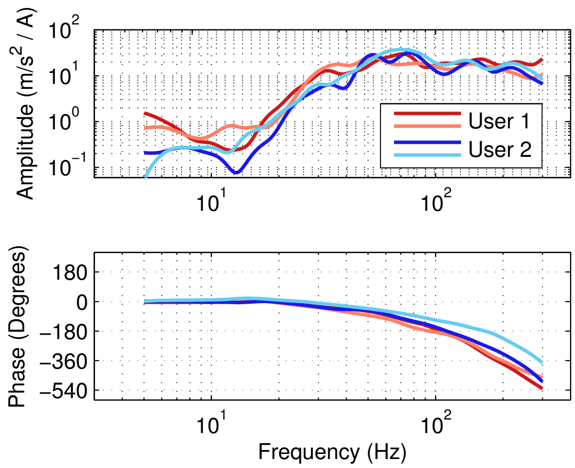

Fig. 8. The frequency response of the hand-pen-actuator system. The response signal was shifted 32 milliseconds in time to account for time-delay caused by the computer soundcard's output buffer.

Fig. 9. The Wacom tablet (center) displaying the haptic texture user-study software, and the real texture mounted alongside it (right).

5 User Study

After finishing the hardware and software of our texture system, we wanted to understand how it would be perceived by users. Specifically, we were interested in the vibration strength that users would prefer, and the level of realism that the system could achieve. To limit the length of individual subject trials and avoid subject fatigue, the study used only six of the eight materials that were modeled; cardboard and brushed plastic were omitted because they are somewhat similar to other textures.

5.1 Experiment Setup

All subjects sat at a desk with the tablet in front of them, and they wore a set of headphones playing pink noise to mask ambient sounds. To enable the subject to directly compare virtual and real textures, the tablet was modified so that real texture samples could be secured to the right half of the surface (Fig. 9). On one side of the tablet was a numeric keypad that the subject used for data entry.

5.2 Experiment Procedure

The study employed a forced-choice, one-up one-down adaptive staircase method [37] to determine the vibration strength preference of each user for each texture. During each trial two surfaces were presented to the participant, one real surface and one virtual surface (consisting of vibrations generated using an LPC model as described in Section 4.1.2), with an adjustable value for γ. A value of $\gamma = 1$ corresponds to the value we expected subjects to prefer, if our texture modeling and rendering system performed as desired.

Prior to the beginning of the experiment, the subject gave informed consent and filled out a demographic

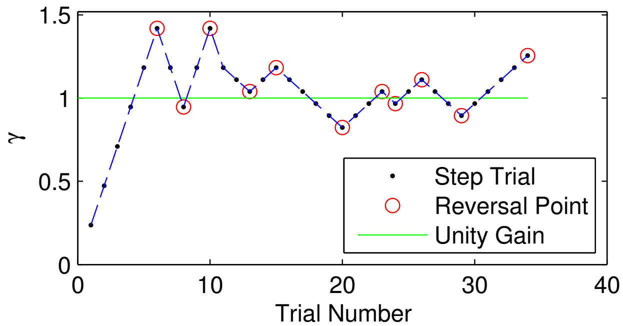

Fig. 10. A sample trial of the staircase method used to find the average preferred value of γ.

questionnaire. Then the subject was instructed that their task was to determine which surface, real or virtual, felt as though it had stronger vibrations. The subject was to enter this decision on the keypad by pushing one of two keys labeled 'real' and 'virtual'. The subject was instructed to use similar motions and forces on the two samples, to take as much time as necessary to satisfactorily compare the two surfaces, and to feel free to switch between the real and virtual surfaces as often as desired. After the goal of the experiment was clear, the subject was asked to put on the headphones and grasp the stylus in any way they felt comfortable, while avoiding placing any part of the hand on the voice-coil actuators. The experiment was initiated when the experimenter attached one of the six surface samples onto the Wacom tablet.

After each entry, the value of γ was automatically adjusted, and the subject repeated the comparison. Each sequence began with $\gamma = 0.237$, a value that caused the virtual surface to feel significantly weaker than the real surface. The initial step size for our staircase method was set to 0.237, and γ was incremented each time the subject rated the real surface to be stronger than the virtual surface. γ was decremented each time the real surface was rated weaker than the virtual surface. The step size was decreased to 0.072 after the first three response reversals (a reversal was considered to have occurred when the slope of the response changed sign). The sequence was terminated after eight reversals at the 0.072 step size level, for a total of eleven reversals, as seen in Fig. 10. The step size values chosen for this study were determined from the range and mean values of γ obtained in a pilot study that used the method of adjustments.

After each staircase sequence was completed, the participant was asked to rate the realism of the virtual surface using four different values for γ. For this part of the study, the experimenter first averaged the final eight reversal points found in the previously completed staircase trial. Next, the strength of the virtual surface was set to one of four possible levels: 0%, 50%, 100%, or 150% of this average γ. Finally, the subject was asked to freely explore the virtual and real surfaces while judging how well the virtual surface was able to match the sensations of the real surface. The subject then rated the quality of the match on a scale from 0 to 100, with a value of 0 representing that the two surface feel 'nothing alike at all,' and a value of 100 meaning that the two surfaces were 'completely indistinguishable.' It was explained to the subject that this comparison was more qualitative than the vibration magnitude comparisons they were asked to make during the staircase trials.

The experiment was continued in this manner for five new materials, giving six materials total. The order in which the materials were presented was selected randomly by the experimenter beforehand. Visual indicators were provided to the subject to keep them below the maximum force and velocity levels of the LPC model during all interactions with both the real and virtual materials. If the subject exceeded the maximum force level of the LPC model, the tablet background turned red. If they exceeded the maximum speed, the background turned yellow. And if both were exceeded, the background turned orange. Subjects were instructed to keep their force and speed below these levels. A total of nine subjects participated in the study (five male and four female). All subjects were between the ages of 21 and 30. One subject was left handed, and all completed the experiment with their dominant hand.

5.3 Results

This section analyzes the vibration gains and realism ratings provided by subjects during the experiment. We use $\alpha = 0.05$ except where noted.

5.3.1 Vibration Gain

Eleven γ reversals were recorded for each subject for each of the six materials. The first three reversals were discarded, and the final eight were averaged together to obtain a single value $\widehat{\gamma}$, for a total of 54 values (9 subjects $\times$ 6 materials). A two-way analysis of variance (ANOVA) was performed using $\widehat{\gamma}$ as the response variable. Subject and material were set as explanatory variables and treated as random effects. This analysis showed that the material type did not significantly affect the value of γ selected by subjects ($F(5{,}53) = 1.36$, $p = 0.26$). Subject did have a significant effect on γ ($F(8{,}53) = 3.25$, $p = 0.006$, $\eta^2 = 0.3573$). Subsequent post-hoc t-tests were run comparing all subjects, adjusted using a Bonferroni correction. These tests showed that only the mean of subject 3 differed significantly from subjects 1, 4, and 6, using $\alpha = 0.05/36 = 0.0014$ ($p < 0.001$ for all three comparisons). The means of all other subjects showed no significant differences from one another. A mean value of $\gamma = 0.9 \pm 0.15$ was calculated by averaging $\widehat{\gamma}$ over all subjects and all textures.

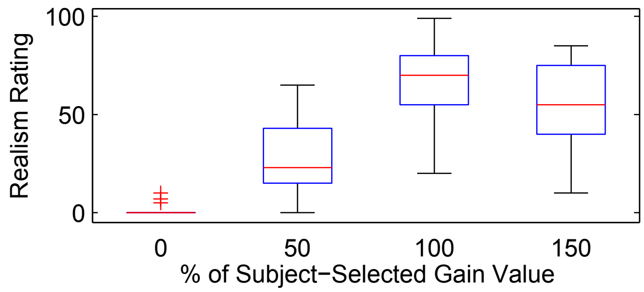

Fig. 11. Realism ratings given by subjects for percentages of the subject-selected γ value. All groups were statistically significant from one another, with the 100% gain level being rated the most realistic.

5.3.2 *Texture Realism*

216 realism ratings were recorded during the study (9 subjects $\times$ 6 materials $\times$ 4 γ levels). A three-way ANOVA was performed to understand how the value of γ affected realism rating. Realism rating was treated as the response variable, and subject, material, and gain level were treated as the explanatory variables. Both subject and material were treated as random effects. This analysis showed that the γ level significantly affects the realism rating selected by subjects (F(3,215) = 242, $p < 0.001$, η^2 = 0.705). Subsequent post-hoc t-tests, adjusted using a Bonferroni correction, showed that all four γ levels were significantly different from one another using $\alpha = 0.05/6 = 0.0083$ ($p < 0.001$ for all except the 100% and 150% comparison, for which p = 0.0081). As seen in Fig. 11, the 100% gain level resulted in the highest realism ratings, followed by 150%, 50%, and then 0%. The box edges indicate the 25% to 75% percentiles, the whiskers include the range of points within 1.5 times the interquartile range, and all outlier points beyond the whisker range are indicated by a +. The middle line indicates the median value.

A two-way ANOVA was then performed using the realism ratings recorded during only the γ = 100% trails. Realism rating was treated as the response variable, and subject and material were treated as the explanatory variables, both as random effects. Material was found to have a significant effect on realism rating at (F(5,53) = 6.49, $p < 0.001$, η^2 = 0.30). Post-hoc t-tests with a Bonferroni correction resulted in a total of four pairwise comparisons with significant differences (out of 15 total comparisons), as seen in Fig. 12. Box plot properties are the same as described for Fig. 11. Subject was also found to have a significant effect on realism rating (F(8,53) = 4.49, $p < 0.001$, η^2 = 0.33). Post-hoc t-tests with a Bonferroni correction resulted in a total of six pairwise comparisons with significant differences (out of 36 total comparisons). The mean realism rating averaged over all materials and subjects at the 100% gain level was 65.4, with a standard deviation of 19.0.

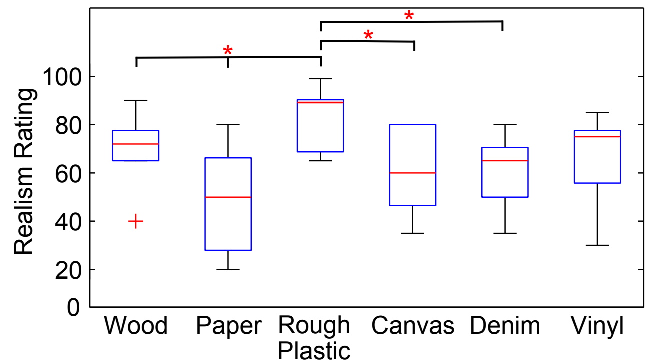

Fig. 12. Realism ratings given by subjects for different types of material. Statistically significant differences are marked above with a bar and * symbol.

5.4 Discussion

The user study provided a variety of interesting findings about the vibration gain γ and the realism of our virtual textures. We found that subjects tended to prefer values of γ very near the unity gain value, and that this preference was not affected by material. Subject was found to have a slight effect on γ preference, with one out of nine subjects differing from three others. We suspect this difference stems from variations in hand mass and grip style, as well as personal preference, but it is a complicated issue that will require further study. In practice, we have found it best to set γ to unity and to finely adjust it to each user's preference, if necessary.

Our study of realism rating showed that the subject's chosen value of γ leads to the highest ratings of perceived realism (Fig. 11). This shows that values of γ that generate accelerations at the subject's hand that are roughly equivalent in magnitude to the accelerations experienced during real texture interaction result in the most realistic virtual sensations. This result contrasts to some of our previously reported results in [30], where subjects tended to prefer a gain level higher than the unity value. In this previous research only a single axis of vibration was relayed to the user, as opposed to capturing the power of all three axis with the DFT321 algorithm. The preference of the unity gain value in our new study now indicates that the proper level of vibration magnitude is being conserved in our DFT321/LPC approach.

The ANOVA on realism ratings showed that material type is a significant factor. Rough plastic received the highest ratings, while paper received the lowest. Subjects tended to vary in their ranking of different materials, resulting in large standard deviations for each virtual material's realism. It is not yet clear exactly what features cause our virtual materials to feel unrealistic, with some subjects commenting on a discrepancy in kinetic friction, and others citing a perceived difference in vibrations. However, it appears that a trend may exist where rougher surfaces with more vibrations (such as rough plastic and vinyl) may be perceived as more realistic than smoother surfaces (such as paper).

6 Conclusion

This paper presents methods for creating the most detailed and realistic haptic virtual textures to date. We began by constructing a new recording system that captures motion, force, and acceleration data while a user interacts with textured surfaces. We presented a method for reducing the recorded three-dimensional acceleration information down to a one-dimensional signal, and we detailed a process for converting each one-dimensional time-domain signal into a frequency-domain vibration model. We store a set of such models at different speeds and forces as a representation of how that surface feels with that tool.

We also presented a real-time method for recreating unique time-domain signals from our frequency-domain models, based on the tool's speed and force. We describe novel hardware that is capable of rendering these high-bandwidth signals to the user via voice coil actuators. Finally, we tested our system in a user study that involved direct real-to-virtual comparisons. The average vibration gain chosen by subjects was near the gain level that would produce one-to-one vibration output from our model, and the average realism rating of 65.4/100 substantiates the merits of our approach. This system has also been positively received during demonstrations at Haptics Symposium 2010, ICRA 2010, and EuroHaptics 2010.

While our approach largely succeeds at recreating the feel of real textured surfaces, there are several areas that we are continuing to investigate with future research. Our current modeling method requires long streams of data at constant speed and force values, which takes time and skill to collect; we are developing methods for building models from less controlled interaction data. Our models currently lack any directional information or specific location-based information, as would be needed to render corduroy fabric or strongly grained wood with knots. We intend to develop more advanced texture modeling and storage methods that are capable of quickly retrieving data from higher-order multi-dimensional datasets. Lastly, we are working to port this texture-rendering approach to the open-source haptics package CHAI 3D, so that it can be used with three-dimensional virtual objects and more traditional haptic devices.

By outlining the theoretical basis of our algorithms and the practical implementation of our system, we hope this approach can become a widely adopted tool in the haptics community. We believe that combining our texture approach with other state-of-the-art haptic rendering methods for properties such as object shape and friction will yield virtual simulations that achieve superior overall realism results.

Acknowledgments

The authors thank Nils Landin for his work on the DFT321 signal synthesis techniques, William McMahan for his research on voice coil actuation approaches, and Craig McDonald for his efforts in designing and running our pilot user study. This material is based upon work supported by the National Science Foundation under Grant No. 0845670.

References

[1] S. J. Lederman, "Tactile rougness of grooved surfaces: The touching process and effects of macro- and microsurface structure," *Perception & Psychophysics*, vol. 16, no. 2, pp. 385–395, 1974.

[2] G. Campion and V. Hayward, "Fundamental limits in the rendering of virtual haptic textures," in *Proceedings IEEE World Haptics*, Washington, DC, USA, 2005, pp. 263–270.

[3] K. J. Kuchenbecker, J. P. Fiene, and G. Niemeyer, "Improving contact realism through event-based haptic feedback," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 219–230, March/April 2006.

[4] M. C. Çavuşoğlu, D. Feygin, and F. Tendick, "A critical study of the mechanical and electrical properties of the PHANToM haptic interface and improvements for high-performance control," *Presence: Teleoperators and Virtual Environments*, vol. 11, pp. 555–568, 2002.

[5] M. A. Otaduy and M. C. Lin, "Rendering of textured objects," in *Haptic Rendering: Foundations, Algorithms, and Applications*, M. Lin and M. Otaduy, Eds. AK Peters, 2008, ch. 18, pp. 371–393.

[6] S. J. Lederman and R. L. Klatzky, "Extracting object properties through haptic exploration," *Acta Psychologica*, vol. 84, pp. 29 – 40, 1993.

[7] R. Klatzky and S. J. Lederman, "Perceiving object properties through a rigid link," in *Haptic Rendering: Foundations, Algorithms, and Applications*, M. C. Lin and M. Otaduy, Eds. AK Peters, 2008, ch. 1, pp. 7–19.

[8] S. Bensmaïa, M. Hollins, and J. Yau, "Vibrotactile intensity and frequency information in the Pacinian system: A psychophysical model." *Perception and Psychophysics*, vol. 67, no. 5, pp. 828–841, 2005.

[9] T. Yoshioka, S. J. Bensmaïa, J. C. Craig, and S. S. Hsiao, "Texture perception through direct and indirect touch: An analysis of perceptual space for tactile textures in two modes of exploration." *Somatosensory and Motor Research*, vol. 24, pp. 53–70, 2007.

[10] K.-U. Kyung and J.-Y. Lee, "Ubi-pen: A haptic interface with texture and vibrotactle display," *IEEE Computer Graphics and Applications*, vol. 29, no. 1, pp. 56–64, 2009.

[11] J. C. Lee, P. H. Dietz, D. Leigh, W. S. Yerazunis, and S. E. Hudson, "Haptic pen: a tactile feedback stylus for touch screens," in *Proceedings of the ACM symposium on User interface software and technology*, 2004.

[12] S. A. Wall and W. S. Harwin, "Effects of physical bandwidth on perception of virtual gratings," in *Proceedings of the ASME Dynamics Systems and Control Divsion*, vol. 69, 2000, pp. 1033–1039.

[13] D. A. Kontarinis and R. D. Howe, "Tactile display of vibratory information in teleoperation and virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 4, pp. 387–402, 1995.

[14] C. Basdogan, C.-H. Ho, and M. A. Srinivasan, "A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments," in *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 61, 1997, pp. 77–84.

[15] S. Choi and H. Z. Tan, "Toward realistic haptic rendering of surface textures," in *IEEE Computer Graphics & Applications (Special Issue on Haptic Rendering - Beyond Visual Computing)*, 2004, pp. 40–47.

[16] M. A. Costa and M. R. Cutkosky, "Roughness perception of haptically displayed fractal surfaces," in *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 69, 2000, pp. 1073–1079.

[17] A. Crossan, J. Williamson, and R. Murray-Smith, "Haptic granular snythesis: Targeting, visualisation and texturing," in *Proceedings of the Eighth International Conference on Information Visualisation*, 2004, pp. 527–532.

[18] S. Andrews and J. Lang, "Haptic texturing based on real-world samples," in *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE)*, 2007.

[19] S. A. Wall and W. S. Harwin, "Modelling of surface identifying characteristics using fourier series," in *Proceedings of the ASME Dynamic Systems and Control Dvisison*, 1999, pp. 65–71.

[20] M. Minsky, O.-y. Ming, O. Steele, F. P. Brooks, Jr., and M. Behensky, "Feeling and seeing: issues in force display," in *SI3D '90: Proceedings of the 1990 Symposium on Interactive 3D graphics*, vol. 24, no. 2. ACM, 1990, pp. 235–241.

[21] M. Minsky and S. J. Lederman, "Simulated haptic textures: Roughness," in *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 58. ASME, 1996, pp. 421–426.

[22] J. Siira and D. K. Pai, "Haptic texturing - a stochastic approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 1996, pp. 557–562.

[23] J. P. Fritz and K. E. Barner, "Stochastic models for haptic texture," in *Telemanipulator and Telepresence Technologies III*, M. R. Stein, Ed., vol. 2901(1). SPIE, 1996, pp. 34–44.

[24] A. M. Okamura, R. J. Webster, J. Nolin, K. W. Johnson, and H. Jafry, "The haptic scissors: Cutting in virtual environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, September 2003, pp. 828–833.

[25] K. E. MacLean, "The "Haptic Camera": A technique for characterizing and playing back haptic properties of real environments," in *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 58, 1996, pp. 459–467.

[26] D. Green and J. K. Salisbury, "Texture sensing and simulation using the PHANToM: Towards remote sensing of soil properties," in *Proceedings of the Second PHANToM Users Group Workshop (PUG)*, 1997, pp. 19–22.

[27] J. M. Romano, T. Yoshioka, and K. J. Kuchenbecker, "Automatic filter design for synthesis of haptic textures from recorded acceleration data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[28] V. L. Guruswamy, J. Lang, and W. S. Lee, "Modelling of haptic vibration textures with infinite-impulse-response filters," in *Proceedings of IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, November 2009, pp. 105–110.

[29] D. K. Pai, K. v. d. Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau, "Scanning physical interaction behavior of 3D objects," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 87–96.

[30] W. McMahan, J. M. Romano, A. M. A. Rahuman, and K. J. Kuchenbecker, "High frequency acceleration feedback significantly increases the realism of haptically rendered textured surfaces," in *Proceedings of the IEEE Haptics Symposium*, March 2010, pp. 141–148.

[31] "The Linux Wacom Project," April 2010. [Online]. Available: http://linuxwacom.sourceforge.net/

[32] J. Bell, S. Bolanowski, and M. Holmes, "The structure and function of Pacinian corpuscles: A review." *Progress in Neurobiology*, vol. 42, no. 1, pp. 79–128, 1994.

[33] N. Landin, J. M. Romano, W. McMahan, and K. J. Kuchenbecker, *Haptics: Generating and Perceiving Tangible Sensations, Proc. EuroHaptics, Part II*, ser. Lecture Notes in Computer Science. Springer, July 2010, vol. 6192, ch. Dimensional Reduction of High-Frequency Accelerations for Haptic Rendering, pp. 79–86.

[34] K. J. Kuchenbecker, J. M. Romano, and W. McMahan, "Haptography: Capturing and recreating the rich feel of real surfaces," in *Proceedings of the 14th International Symposium of Robotics Research (ISRR)*, Lucerne, Switzerland, 2009.

[35] "PortAudio - Cross-Platform Audio API," April 2010. [Online]. Available: http://www.portaudio.com/

[36] W. McMahan and K. J. Kuchenbecker, "Haptic display of realistic tool contact via dynamically compensated control of a dedicated actuator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent RObots and Systems*, 2009.

[37] H. Levitt, "Transfored up-down methods in psychoacoustics," *The Journal of the Acoustical Society of America*, vol. 49, pp. 467–477, 1971.

Joseph M. Romano (S'07) is a Ph.D. student in the department of Mechanical Engineering and Applied Mechanics (MEAM) at the University of Pennsylvania. Joe works in the Haptics Group, a part of the GRASP Laboratory, investigating the relationship between kinesthetic large-scale motion information and localized tactile and cutaneous signals to improve human and machine haptic systems. He completed a Masters in MEAM at the University of Pennsylvania in May of 2010, and B.S. degrees in both Computer Science and Engineering Mechanics at the Johns Hopkins University in 2007.

Katherine J. Kuchenbecker (S'04-M'06) is the Skirkanich Assistant Professor of Innovation in Mechanical Engineering and Applied Mechanics at the University of Pennsylvania. Her research centers on the design and control of haptic interfaces, and she directs the Penn Haptics Group, which is part of the GRASP Robotics Lab. Dr. Kuchenbecker serves on the program committee for the IEEE Haptics Symposium, and she has won several awards for her research, including an NSF CAREER award in 2009 and Popular Science Magazine Brilliant 10 in 2010. Prior to becoming a professor, she was a postdoctoral fellow at the Johns Hopkins University, and she earned her Ph.D. in Mechanical Engineering from Stanford University in 2006.