

Received July 13, 2020, accepted August 4, 2020, date of publication August 11, 2020, date of current version August 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015861

Hybrid Framework for Haptic Texture Modeling and Rendering

SUNGHWAN SHIN¹, (Student Member, IEEE), AND
SEUNGMOON CHOI¹, (Senior Member, IEEE)

Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

Corresponding author: Seungmoon Choi (choism@postech.ac.kr)

This work was supported in part by grants No. 2017-0-00179 (HD Haptic Technology for Hyper Reality Contents) and No. 2020-0-00594 (Morphable Haptic Controller for Manipulating VR · AR Contents) from the Institute of Information & Communications Technology Planning & Evaluation (IITP) and a grant NRF-2017R1A2B4008144 from National Research Foundation of Korea (NRF), all funded by the Korean government.

ABSTRACT We present a hybrid framework for the modeling and rendering of both homogeneous and inhomogeneous textures with high realism by combining force and vibration feedback. Given a real texture sample, we first capture a force-feedback model using photometric stereo, which is one of the most accurate algorithms for 3D surface reconstruction in computer vision. This method allows us to model the micro-geometry of the surface and represent it by a two-dimensional (2D) height map in a very high resolution to the order of $10\ \mu\text{m}$. We also provide a new force rendering algorithm adequate for such an accurate and fine texture model. By a user study, we verify that the proposed force modeling and rendering algorithms are suitable for our hybrid texture framework. Second, we model the homogeneous and stationary vibrational characteristics of the real texture sample using the prevalent LPC (linear predictive coding)-based method. Third, we merge the force and vibration feedback models appropriately into a hybrid framework on the basis of their spectral characteristics. Last, another user study that assessed the perceived similarity between real textures and their virtual simulations demonstrates that our hybrid framework can achieve a high level of realism. To our knowledge, our framework is unique in that it can model inhomogeneous (in addition to homogeneous) real textures and render them in a virtual environment with high realism by providing both force and vibrational sensory cues.

INDEX TERMS Haptics, texture, force feedback, vibration feedback, hybrid, perception, realism.

I. INTRODUCTION

In haptics research, texture is one of the most important properties. Haptic texture generally refers to the feel experienced when touching the fine structural irregularities on the surface of an object. As texture greatly impacts the user's haptic experiences, modeling and rendering a haptic sensation identical to that of the real texture have been one of the major research topics in haptics for almost two decades.

However, it is nearly impossible to formulate the physical process of generating stimuli for haptic texture because of the complexity of the physics involved. Consequently, researchers have developed different types of approximation model, such as geometry-based deterministic force models [1]–[5], stochastic models [6]–[8], and data-driven acceleration models [9]–[14]. Despite such focused research

endeavors, we are still void of standard methods for modeling and rendering realistic haptic textures similar to real textures. It is largely due to the fact that haptic texture perception consists of multiple perceptual dimensions, such as roughness-smoothness, warmth-coldness, hardness-softness, and stickiness-slipperiness [15]. Furthermore, it remains very challenging to model and render *inhomogeneous* texture in which the textural property depends on the contact position.

For homogeneous texture, a data-driven approach that captures and renders vibrational contact acceleration based on Linear Predictive Coding (LPC; adapted from technologies for speech) is the current state of the art [12], [16], [17]. This method aims to create a vibration profile that has the frequency spectrum identical to that of the real contact acceleration that occurs when the user scans the textured object using a rigid tool held in her/his hand. By doing so, this method effectively simulates the sensation of roughness

The associate editor coordinating the review of this manuscript and approving it for publication was Pengcheng Liu¹.

of the target object. Several studies demonstrated that the LPC-based method is appropriate for the haptic modeling and rendering of homogeneous texture.

A problem is that extending vibration-based data-driven methods to inhomogeneous texture is not straightforward. In this case, contact acceleration data must be collected and modeled locally at every position, or at least in a very dense array of surface points. It increases the amount of real contact data exponentially and also the size and complexity of a data-driven model commensurately.

To solve this problem, we propose a hybrid haptic texture modeling and rendering framework using force and vibration feedback in which both feedback models are captured from real textured surfaces. Our framework is consistent with the duplex theory of tactile texture perception [18]. The theory indicates that tactile texture perception is mediated by a combination of vibrational cues for fine texture and spatial cues for coarse texture. Inspired by this theory, we break the problem of modeling inhomogeneous texture into two steps: one for a coarse inhomogeneous component of the texture, and the other for a fine homogeneous vibrational component. The former is rendered by force feedback, and the latter is by vibration feedback.

A. RELATED WORK

For haptic texture rendering, most early algorithms used geometry profiles for force feedback based on the two seminal studies by Minsky [1] and Massie [3]. In 1995, Minsky showed that rendering only the lateral force computed from the spatial gradient of a texture height map can elicit different roughness sensations. A year later, Massie demonstrated that texture sensations can be invoked by varying the magnitude of normal force as the user strokes a textured surface. These findings have inspired the designs of many texture rendering algorithms using force (kinesthetic) feedback, e.g., [4], [5], [19]–[22]. These force-feedback texture rendering algorithms are generally adequate for rough textures, even those including visible geometric features.

In contrast, haptic modeling methods for the geometric textural profiles of real surfaces have not been actively studied, although it is the first step to re-create real textures in a virtual environment. The methodology for texture geometry capture generally relies on a profilometer (or a similar device). For example, Costa *et al.* used an optical profilometer to scan rock surfaces [23], and Wall *et al.* used a linear variable differential transformer to measure the displacement of a probe moving on a surface [24]. Pai *et al.* developed a wireless haptic texture sensor (WHaT) equipped with a 3D accelerometer, a 1D force sensor, and a visual marker [25]. Using a different approach, Ikei *et al.* relied on photography to retrieve the height map of a textured surface [26]. This image-based approach is convenient and greatly faster than the mechanical scanning methods, but it has not been popular due to the low modeling resolution and accuracy of the early apparatus and algorithms.

Vibration is also an important feature in haptic texture perception. Okamura *et al.* proposed a texture rendering

algorithm based on measured vibrations [9]. In this method, force rendering for a virtual surface was added by a vibration in the form of decaying sinusoidal pulses to simulate the exploration of real surfaces, such as tapping and stroking with a rigid stylus. Since then, manual surface exploration using a handheld sensorized stylus has been the dominant data collection method for data-driven texture modeling. This is inexpensive and easy to use, and also allows freeform scanning of surface. Using WHaT [25], Lang and colleagues presented a series of studies for data-driven haptic modeling and rendering in virtual environments [10], [27], [28]. Another successful approach proposed by Kuchenbecker and colleagues is based on LPC, and their method generates virtual textures with a very high level of realism in terms of the roughness of isotropic micro-textures [29]. They consistently improved their methods [12], [16], [17] and opened a public repository of one hundred vibration-based texture models [30]. While its application is limited to homogeneous textures, the LPC-based method can simulate quite realistic virtual textures. Recently, Jeon and his group extended the LPC-based model to anisotropic texture by using a two-dimensional velocity vector as an additional input feature [13]. Contact acceleration data is collected and segmented based on the velocity vector. Their method effectively handles vibrations induced by anisotropic textures.

B. OUR APPROACH

We call our approach ‘hybrid’ because it integrates both force and vibration feedback to deliver more realistic texture experiences. According to the duplex theory of texture perception [31], a haptic texture sensation is a combination of two signals transferred through two neural channels. Fine textural features (e.g., particle size $< 100 \mu\text{m}$) induce contact vibration when the user scans the textured surface. The Pacinian (PC) mechanoreceptive channel mediates this contact vibration in the form of a temporal signal. Conversely, coarse textural features (e.g., particle size $\geq 100 \mu\text{m}$) at bare finger contact are encoded in terms of their spatial layout through the SA (slowly adapting) 1 channel that has Merkel receptors as the end organs. In the tool-mediated exploration, the user can perceive the coarse texture information by temporally integrating the force cues generated during scanning. This theory inspired us to develop and render two respective models for more realistic haptic texture rendering.

To model the potentially inhomogeneous micro-geometry of a real surface, we rely on photometric stereo, where the surface geometry is estimated by analyzing the correspondence among different images photographed using a fixed camera under precisely controlled lighting conditions. Photometric stereo can capture high-frequency tiny variations on the target surface. Since the boundary between coarse and fine textural features is approximately $100 \mu\text{m}$ [31], such an accurate algorithm is still required for the modeling of coarse textures. A 3D profilometer, although extremely accurate, is not attractive for our purpose due to its high price and slow scanning velocity (in the order of 10^{-3} mm/s). We designed

and built a custom apparatus to apply photometric stereo to texture modeling. A new force-feedback rendering algorithm for our high-resolution geometry is proposed as well.

To model the contact vibration of texture scanning, we adopt the LPC-based method. As described earlier, this method is the state of the art for modeling homogeneous haptic textures induced by fine textural features using contact vibration signal that exhibits stationarity. To extend this method to inhomogeneous textures, we filter out the non-stationary components of the vibration signal induced by textural inhomogeneity. This non-stationary component is included in the micro-geometry model that we capture by photometric stereo.

Then we integrate the two models to recreate virtual textures with high resolution and realism. Our hybrid framework was evaluated by two user studies: one for the rendering performance of our force-feedback algorithm and the other for the realism achieved by our framework.

The main contributions of our work are: 1) the combination of an image model and a vibration model for haptic texture modeling, 2) the integrated texture rendering algorithm using both force and vibration feedback, 3) one user study to select the most appropriate force-feedback rendering algorithm for hybrid texture rendering, and 4) the other user study to evaluate our hybrid framework in terms of realism, i.e., the similarity between real and virtual textures.

II. DATA COLLECTION AND MODELING

For force feedback in our hybrid framework, we construct the height map of a real texture material and also estimate its stiffness and friction coefficients. For vibration feedback, we model the contact vibration signal using a finite filter by LPC. Each modeling process is independent for better usability.

Multi-modal modeling and rendering are prone to exaggeration. Both force and vibration signals are captured from a single touch interaction, and they may include common components. We prevent this problem of overcompensation by delineating the two signals at the critical frequency f_c in the frequency domain. Using an iterative process, f_c is set to the maximum frequency that contains the non-stationary components of contact vibration. The frequency bandwidth of force signal is limited to be below f_c , whereas the texture properties above are handled by the vibration signal.

We describe each modeling process in this section.

A. CONTACT VIBRATION MODELING

To model and synthesize contact vibration signals, we use the LPC-based method [29]. LPC is standard in speech processing for its good encoding quality at a low bit rate. LPC expresses a speech signal using a linear filter and enables the reconstruction of a signal that has the same spectrum with the original signal. This advantage is attractive to many applications in which spectral information is important, including haptic texture rendering.

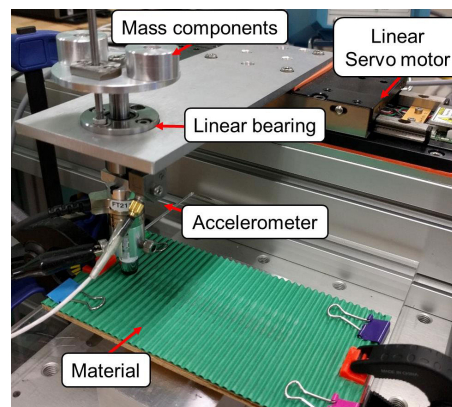


FIGURE 1. A motorized texture scanning device. The vibration induced by the interaction between the stylus and the material is captured during scanning.

LPC assumes that the input signal is stationary and so is applicable to only homogeneous textures. Such a texture results in contact vibration signals of the same statistical properties regardless of contact position or scanning trajectory. However, many real textures exhibit inhomogeneity, and it induces a non-stationary vibration signal. If the target texture is inhomogeneous but anisotropic, we can apply the velocity-based data segmentation method [13] owing to the texture's regularity. The segmented data can be effectively modeled and rendered by an LPC-based method. This idea is difficult to be extended to general inhomogeneous textures because the number of segments and the corresponding models may increase indefinitely. To tackle this problem, we extract only the stationary component of a contact vibration signal using a high-pass filter and model it using LPC. The non-stationary component of the texture is handled by force feedback (Section II-B).

To collect contact acceleration signals, we use a motorized texture scanning device [14]. Manual data collection using a sensor-equipped handheld tool demonstrates good performance for modeling isotropic and homogeneous textures [25]. For inhomogeneous textures, however, it may be very difficult for the user to keep the scanning velocity and normal force constant during data collection. This problem is not the case for our texture scanning device of high accuracy and repeatability.

Figure 1 depicts our texture scanning device. A stylus pen is moved by a linear motor (MX80L, Parker). The normal force is adjusted by changing the mass placed on the shaft. Contact vibration is measured using an accelerometer (8740A500, Kistler). Since the contact acceleration depends on the normal force and the tangential velocity, we collect data for various normal forces (0.85, 1.0, 1.15, and 1.3 N) and tangential velocities (1.5, 3.0, 4.5, \dots , 12 cm/s) in a grid. For each combination of the two variables, we build an LPC model using the collected contact acceleration data.

The collected data is high-pass filtered to remove the non-stationary component originated from the texture inhomogeneity. The cut-off frequency of the high-pass filter is

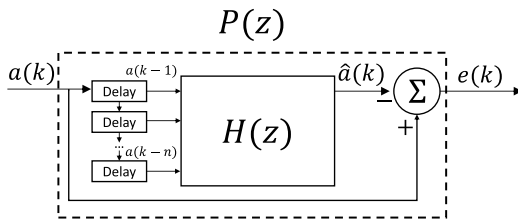


FIGURE 2. A diagram of LPC modeling from [29]. The model $P(z)$ contains a linear transfer function $H(z)$ that takes the acceleration history $\mathbf{a}(k-1)$ to generate $\hat{a}(k)$, the estimate of $a(k)$.

determined by iterative search. We find the smallest cut-off frequency for which the resulting filtered signal passes the augmented Dickey-Fuller test, a widely used test for signal stationarity [32]. The cut-off frequency depends on the texture, varying generally in a 50–150 Hz range.

We use the same LPC-based model proposed by Kuchenbecker's group [29]. The filtered contact acceleration signal is fed to the LPC modeling procedure as shown in Figure 2. The goal is to find a linear filter that predicts the next sample of contact acceleration $a(k)$ from the previous acceleration values. To this end, we need to find an optimal filter vector \mathbf{h} that produces the minimum value of $e(k)$:

$$e(k) = a(k) - \hat{a}(k) = a(k) - \mathbf{h}^T \mathbf{a}(k-1), \quad (1)$$

where $\mathbf{a}(k-1)$ is a vector of the previous contact acceleration samples. We use the vector size of 25 since the performance is saturated above it [29].

For the mean-squared error, \mathbf{h} can be found by solving the Wiener-Hopf equation [33]. In practice, we can use the Levinson-Durbin algorithm [34]. This filter-optimization problem is solved for each combination of normal force and tangential velocity. The resulting filters are used for the LPC texture model. More details can be found in [29].

B. GEOMETRY MODELING

Texture sensations for a virtual surface can be rendered by perturbing the magnitude or direction of the surface normal force. Therefore, the texture quality rendered by a force-feedback device is highly correlated with the accuracy and resolution of the texture height map. We use photometric stereo to model the micro-geometry of a real surface.¹

One of the most important requirements for photometric stereo is the precise control of lighting condition. The most common hardware for this is a lighting dome with multiple light sources installed at fixed locations. Generally, a lighting dome is a dedicated and expensive apparatus covering a large workspace for an object of arbitrary shape. Instead, we designed and custom-built a small lighting dome shown in Figure 3, suitable for relatively small and flat texture

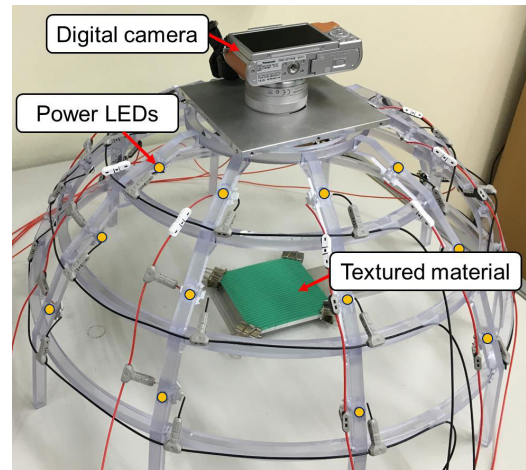


FIGURE 3. Apparatus for texture modeling. The LEDs (marked by orange circles) are installed inside the polycarbonate dome.

samples. We also use an ordinary digital camera for better reproducibility of the system.

Our lighting-dome has a diameter of 15 cm, which is sufficient to cover most haptic texture samples. Thirty power LEDs are installed at specific locations based on the relationship between illumination angle and reconstruction error. According to an empirical experiment [36], the modeling error of photometric stereo is the lowest when the illumination angle from the lighting sources onto the target object is 55° for general objects. The similar holds for the illumination angle of 40° and 70° . Therefore, we place light sources at three elevations of 40° , 55° , and 70° . This configuration is repeated for every 36° in azimuth, resulting in 30 different lighting conditions. Each power LED (1 W) is controlled by a driver circuit to provide sufficient and equivalent illumination.

To construct a height map, we use a photometric stereo algorithm in [37]. This method works well with a regular DSLR camera and multiple flashlights.

Before constructing a height map, we need to estimate the radiance (incident light intensity) function $L(x, y)$ at position (x, y) on the surface. For this, we use a blank white paper as the surface material. This supports the assumption that the normal vector and the albedo (reflection coefficient) are constant over the surface. Therefore, a photographed image of the paper represents the radiance function.

Using the radiance function $L(x, y)$, the photographed light intensity $I(x, y)$ can be represented by

$$I(x, y) = \beta(x, y)L(x, y), \quad (2)$$

where $\beta(x, y)$ is a bidirectional reflectance distribution function (BRDF). The BRDF is a function that defines light reflectance and depends on the surface parameters such as albedo, normal vector, incident light vector, and view vector. Several distinct models exist for BRDF; we use the Lambertian model as it is effective for perfectly diffuse surfaces²

¹Our early work on using photometric stereo for haptic texture modeling and rendering was presented in the 2018 IEEE Haptics Symposium [35]. The main algorithms of [35] are summarized in Section II-B and II-C, along with some improvements.

²Diffuse reflection occurs on a non-glitter and irregular surface, which is the main target of haptic texture modeling and rendering.

despite its simple form [37]. Specifically,

$$\beta(x, y) = \alpha(x, y) \mathbf{n}(x, y)^T \mathbf{l}(x, y), \quad (3)$$

where α is the albedo, \mathbf{n} is the normal vector, and \mathbf{l} is the incident light vector. \mathbf{l} is determined using the lighting dome. Thus, α and \mathbf{n} are the only unknowns in (2) and (3), and they need to be determined at each point (x, y) .

For robust estimation of α and \mathbf{n} , we use multiple photographs taken under N lighting conditions ($N = 30$). The problem is then reduced to an optimization problem as follows:

$$\begin{aligned} (\alpha(x, y), \mathbf{n}(x, y)) = \operatorname{argmin} \sum_{i=1}^N & |I_i(x, y) - \alpha(x, y) \\ & \times \mathbf{n}(x, y)^T \mathbf{l}_i(x, y) L(x, y)|^2. \end{aligned} \quad (4)$$

To solve this optimization problem, we initialize $\mathbf{n} = (0, 0, 1)$ and then find α using singular value decomposition (SVD). A new \mathbf{n} is then computed with this α using SVD. This procedure is terminated until the changes in \mathbf{n} and α become negligible. In most cases, 20 iterations are sufficient to achieve convergence.

Finally, the normal vectors $\mathbf{n}(x, y)$ are integrated over the surface to construct the height map $h(x, y)$ [38].

Detailed results of modeling real texture geometry using the above lighting dome and photometric stereo algorithm are provided in [35]. An analysis of the results demonstrated that the modeling resolution is in the order of $10 \mu\text{m}$.

C. STIFFNESS AND FRICTION MODELING

The stiffness and friction of a material are another major haptic properties, and their behaviors should also be acquired from real materials [39]. In our hybrid texture framework, we use the Hunt-Crossley model for stiffness and the Dahl friction model for friction. We use these relatively simple physics-based models because texture samples are usually thin and allow only shallow deformation. More complex and realistic data-driven methods for general objects would be an overkill for our purpose.

For data collection, we use our texture scanning device with some modifications. To model stiffness, we need profiles of reaction force vs. normal deformation measured from real materials. As shown in Figure 4, we add a linear variable differential transformer (LVDT; LDI-119, Omega Engineering) to the stylus for accurate position sensing in the normal direction and a small, light, accurate triaxial force sensor (Nano 17, ATI Technology) for reaction force sensing. For measurement, we lift the stylus slightly above a texture sample and then drop it. This procedure is sufficient to capture the data of shallow deformations. This is an improvement over the measurement method reported in our previous work [35], leading to better handling of thin texture samples.

To collect friction-related data, such as tangential position, velocity, and force, we use the same hardware as in Figure 1 with the force sensor. We repeat the stop-and-move action

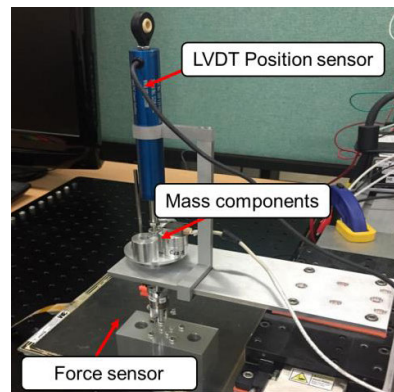


FIGURE 4. An LVDT is attached to the texture scanning device to collect force profiles for stiffness estimation.

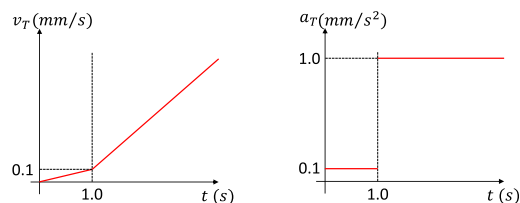


FIGURE 5. Tangential velocity and acceleration profiles used to collect friction-related data.

of the stylus to collect data in the pre-sliding and sliding regimes. Generally, the pre-sliding distance is limited to several micrometers. To acquire sufficient data in the pre-sliding regime, we limit the acceleration and velocity in the tangential direction to be very low for the first few seconds. After that, we increase the acceleration to measure data samples at various tangential velocities (Figure 5). This profile contains all the frictional states, supporting the estimation of all parameters in the Dahl model.

The Hunt-Crossley model has the form

$$F_{HC}(t) = Kz_N^n(t) + Bz_N^n(t)\dot{z}_N(t), \quad (5)$$

where K and B are the stiffness and damping parameters of the object, $z_N(t)$ is the normal displacement, and n is a constant exponent between 1 and 2. This model has been successful in capturing the deformation behavior for haptics [40], [41] by virtue of its balanced performance between modeling accuracy and the ease of parameter identification.

For parameter identification, we use the iterative estimation algorithm proposed by Haddadi and Hashtrudi-Zaad [42]. We supply the collected input data to the identification process. As the stylus for data collection can move only in the vertical direction (Figure 4), the response force can be assumed to result solely from the stiffness and damping. In addition, we discard the input samples that have force amplitudes higher than 2 N as we want to model only the deformation of the fine texture geometry near the surface, not the whole deformation of the object. Users usually exert no more than 2 N of normal force when they stroke the surface of an object [43], so we use this value as the cut-off point.

The Dahl friction model has been frequently used in haptics applications. This simple but useful model can portray the

behaviors of various real frictional responses accurately with reasonable complexity. The original form of the Dahl model is given by

$$\frac{dF_D}{dx_T} = \kappa \left(1 - \frac{F_D}{F_C} \text{sgn}(v_T) \right)^\gamma, \quad (6)$$

where F_D is the Dahl friction force, x_T is the tangential displacement, v_T is the tangential velocity, κ is the stiffness coefficient, and γ is a constant exponent. F_C is the Coulomb friction force, such that

$$F_C = \mu_k F_{norm}, \quad (7)$$

where μ_k is the Coulomb friction coefficient and F_{norm} is the normal load. For efficient computation, we employ a discrete-time representation of the Dahl model [44], such that

$$F_D(t+1) = F_C(t) \text{sgn}(v_T(t)) + (F_D(t) - F_C(t) \text{sgn}(v_T(t))) \cdot e^{-\frac{\kappa}{F_C(t)} |x_T(t) - x_{T0}|}, \quad (8)$$

where x_{T0} is the initial displacement, which is reset to $x_T(t)$ when $v_T(t)$ becomes zero. As the original Dahl approach focuses on modeling near-static friction, it does not consider the tangential velocity. To account for the effect of dynamic friction, we add a simple viscous term after completing the calculation of the Dahl friction as other researchers do [45]. Then the total friction $F_F(t)$ becomes

$$\begin{aligned} F_F(t) &= \mu_b v_T(t) + F_D(t) \\ &= \mu_b v_T(t) + F_C(t-1) \text{sgn}(v_T(t-1)) \\ &\quad + (F_D(t-1) - F_C(t-1) \text{sgn}(v_T(t-1))) \\ &\quad \cdot e^{-\frac{\kappa}{F_C(t-1)} |x_T(t-1) - x_{T0}|}, \end{aligned} \quad (9)$$

where μ_b is the viscous friction coefficient.

Assuming the target surfaces are flat, the force readings in the recorded force profile are divided into the normal load F_{norm} and the tangential friction force F_F .

We identify the parameters in (9) separately according to the movement state (pre-sliding or sliding), as shown in Algorithm 1. When the tangential velocity of the stylus is below $T_{v1} = 0.5$ mm/s, we consider that the system is in the pre-sliding regime. In this regime, the tangential displacement $|x(t) - x_0|$ is very small, and (9) can be approximated by a linear function of κ and $f_D(t)$ using the Taylor expansion of the exponential term. Then (9) is simplified to

$$\begin{aligned} F_F(t) &\simeq \mu_b v_T(t) + F_C(t-1) \text{sgn}(v_T(t-1)) \\ &\quad + (F_D(t-1) - F_C(t-1) \text{sgn}(v_T(t-1))) \\ &\quad \cdot \left(1 - \frac{\kappa}{F_C(t-1)} |x_T(t-1) - x_{T0}| \right). \end{aligned} \quad (10)$$

If we replace $F_D(t-1)$ with $F_F(t-1) - \mu_b v_T(t-1)$ and $F_C(t-1)$ with $\mu_k F_{norm}(t-1)$, then we can write κ as

$$\kappa = \frac{\mu_k F_{norm}(t-1)}{|x_T(t-1) - x_{T0}|} \cdot \frac{F_F(t) - F_F(t-1) - \mu_b v_T(t) + \mu_b v_T(t-1)}{\text{sgn}(v_T(t-1)) - F_F(t-1) + \mu_b v_T(t-1)}. \quad (11)$$

Algorithm 1 Friction Parameter Estimation Algorithm

```

1: procedure Estimator( $\kappa_0, \mu_{k0}, \mu_{b0}, T_{v1}, T_{v2}, T_C$ )
2:    $\kappa \leftarrow \kappa_0$ 
3:    $\mu_k \leftarrow \mu_{k0}$ 
4:    $\mu_b \leftarrow \mu_{b0}$ 
5:   repeat
6:     read current  $v_T, x_T, x_{T0}, F_{norm}, F_F$ 
7:     if  $v_T \leq T_{v1}$  then
8:        $\hat{\kappa} \leftarrow \text{estK}(v_T, x_T, x_{T0}, F_{norm}, F_F, \mu_k, \mu_b)$ 
9:        $\Delta\kappa \leftarrow \hat{\kappa} - \kappa$ 
10:       $\kappa \leftarrow \hat{\kappa}$ 
11:     else if  $v \geq T_{v2}$  then
12:        $\hat{\mu}_k, \hat{\mu}_b \leftarrow \text{estM}(v_T, x_T, x_{T0}, F_{norm}, F_F, \kappa)$ 
13:        $\Delta\mu_k \leftarrow \hat{\mu}_k - \mu_k$ 
14:        $\Delta\mu_b \leftarrow \hat{\mu}_b - \mu_b$ 
15:        $\mu_k \leftarrow \hat{\mu}_k$ 
16:        $\mu_b \leftarrow \hat{\mu}_b$ 
17:     end if
18:     until  $\Delta\kappa, \Delta\mu_k, \Delta\mu_b < T_C$ 
19:     return  $\kappa, \mu_k, \mu_b$ 
20: end procedure

```

As we can read the values of F_F, F_{norm}, v_T from the force sensor and the optical encoder of the linear motor, we use (11) to return κ in the estK function in Algorithm 1.

In the sliding regime where the tangential velocity exceeds $T_{v2} = 3$ mm/s, the exponential term $e^{-\frac{\kappa}{F_C(t)} |x_T(t) - x_{T0}|}$ in (9) converges rapidly to zero. In this case, the friction force can be approximated by the sum of the Coulomb friction and the viscous friction, as follows:

$$F_F(t) \simeq \mu_b v_T(t) + \mu_k F_{norm}(t-1) \text{sgn}(v_T(t-1)). \quad (12)$$

This equation is included in the estM function in Algorithm 1 to determine μ_k and μ_b using the least squares method.

The algorithm stops the repetition when the changes for all the three parameters fall under $T_C = 10^{-9}$.

III. HYBRID TEXTURE RENDERING

After we construct all the models as described in Section II, we use these models to render a hybrid haptic texture. Our hybrid texture rendering algorithm consists of three steps: one step that generates a force signal, another step that synthesizes a vibration signal, and the last step that adjusts the force and vibration signals so that their combination does not lead to an exaggeration and generates a feel that is most similar to the target real texture. Figure 6 shows the whole processes of our texture rendering framework.

For tool-mediated interaction, we attach a custom aluminum stylus-shaped grip to a force-feedback device (Omega.3 by Force Dimension) as shown in Figure 7). A Haptuator (BM3C) by TactileLab is installed for vibration generation inside the grip. A sharp tooltip Under the grip enables the user to interact with real textures (for user studies) in the same manner for virtual textures.

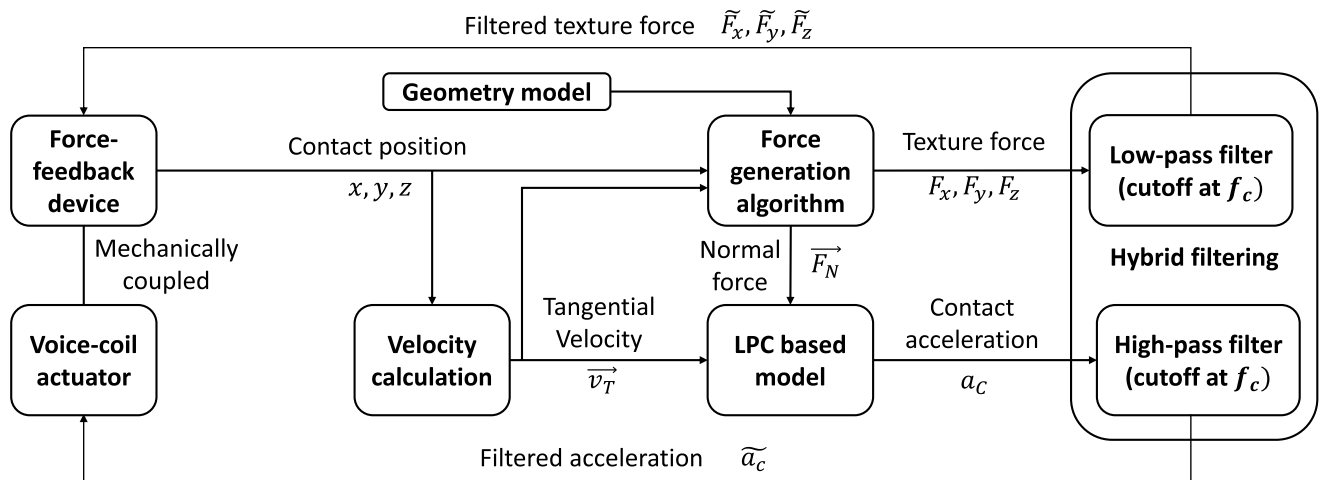


FIGURE 6. Rendering process of the presented framework. First, the contact position information and its derivatives are fed into a force generation algorithm and LPC based model. Our framework uses a force generation algorithm and an LPC-based model simultaneously to synthesize both force-feedback and vibrotactile-feedback signal. The force-feedback signal is low-pass filtered to emphasize the inhomogeneous characteristic of textures. The vibrotactile signal is high-pass filtered to delivers texture perception correlated with high-frequency vibration. The shared cut-off frequency f_c was determined empirically to minimize the cross-talk between two signals from the viewpoint of the duplex theory of texture perception.

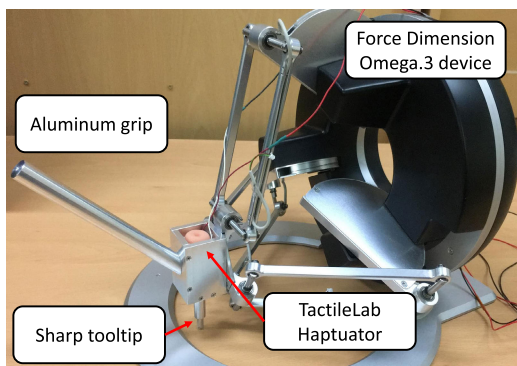


FIGURE 7. Our texture-rendering hardware setup.

Omega.3 device is chosen for its high structural stiffness and spatial resolution. These features improve texture rendering stability, which is a problem in haptic texture rendering using a force-feedback device [5], [21], [46]–[48]. The Haptuator has a flat amplitude response and strong output, which contribute to precise vibrotactile rendering. The voltage command to the Haptuator for a target vibration amplitude is computed using an experimentally-obtained linear calibration function.

A. FORCE GENERATION ALGORITHMS

Generally, force-feedback texture rendering algorithms elicit the rough feel of a virtual surface by perturbing the normal response force. However, they are prone to rendering instability when the normal force fluctuation makes the interaction active in terms of energy flow [46], [47], leading to the occurrence of oscillation or vibration. Such artifacts greatly reduce the realism of virtual exploration. Therefore, a critical requirement in force-feedback texture rendering is to maintain an optimal balance off between realism and stability.

We calculate the texture rendering force as follows. We assume that a flat and thin texture material is overlaid on a stiff base object. First, we calculate the response force \mathbf{F}_O with no consideration of texture:

$$\mathbf{F}_O = \mathbf{F}_M + \mathbf{F}_B + \mathbf{F}_F, \quad (13)$$

where \mathbf{F}_M is the stiffness force from the texture material, \mathbf{F}_B is the stiffness force from the base object under the texture, and \mathbf{F}_F is the friction force. \mathbf{F}_M is computed using the Hunt-Crossley model in (5) using the normal displacement z_N into the material. The base object force \mathbf{F}_B is calculated using a linear spring model. This simple model is sufficient as we do not consider the deformation of the base object. The stiffness coefficient of the base object is set to half the maximum linear stiffness allowed by the force-feedback device. \mathbf{F}_B has the same direction as the surface normal. The friction force \mathbf{F}_F is computed using the Dahl model, and its direction is the opposite of the user's tangential moving direction.

After computing \mathbf{F}_O , we apply texture rendering to generate the final force output \mathbf{F}_I . In Figure 8, we present two representative methods. Algorithm A renders a virtual texture by aligning the direction of the normal force to the gradient of the texture height map [49]. That is,

$$\mathbf{F}_I = |\mathbf{F}_M| \mathbf{n}_t + \mathbf{F}_B + \mathbf{F}_F, \quad (14)$$

where \mathbf{n}_t is the normal vector perpendicular to the textured surface and equivalent to the gradient $\nabla h(x, y)$ of the texture height map $h(x, y)$. As the texture height map varies quickly, algorithm A elicits the sensation of roughness effectively. However, it is quite prone to unstable artifacts [46]–[48].

The next one, algorithm B, is given by

$$\mathbf{F}_I = \mathbf{F}_M + \mathbf{F}_B + (1 - h(x, y))\mathbf{F}_F, \quad (15)$$

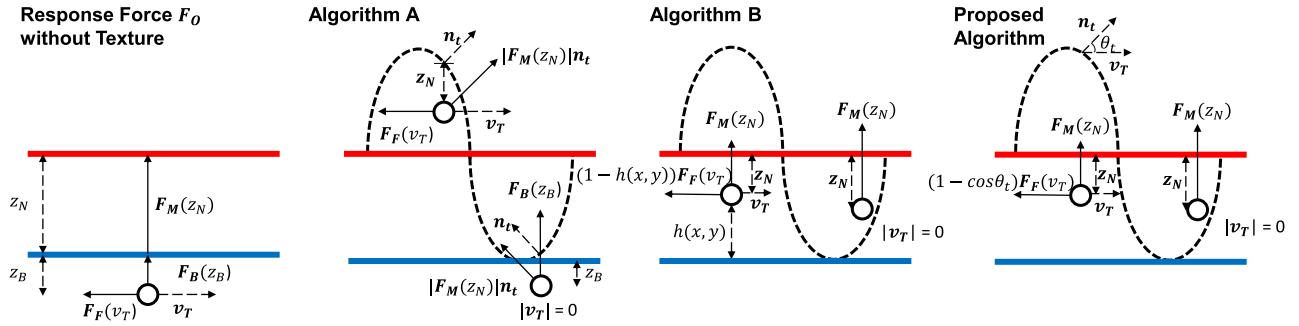


FIGURE 8. Schematic diagrams for texture rendering algorithms with force feedback. Circles represent the position of the haptic tool (HIP: Haptic Interface Point) in the virtual space. Red thick straight lines represent the nominal surface. Blue thick straight lines indicate the border between the texture and the base. Dashed lines show textured surfaces, and $h(x, y)$ is the normalized texture height. z_N and z_B are the normal displacement of the HIP into the nominal surface and the base, respectively. \mathbf{n}_t is the normal vector perpendicular to textured surfaces. \mathbf{v}_T is the tangential velocity of the HIP.

where $h(x, y)$ stands for the normalized texture height. Algorithm B changes the magnitude of the friction force to render a textured surface. It has better rendering stability [5], because the direction of the friction force does not change very rapidly and the magnitude of the texture force falls to zero when the user stops moving.

The two algorithms often face issues when applied to the height maps constructed using photometric stereo. As reported in Section II-B, our photometric stereo method generates height maps with a resolution approximately ten times higher than in prior studies. Our height maps contain greatly smaller and detailed height fluctuations and higher-frequency components than previous height maps. For such height maps, the rendering algorithm A in Figure 8 can be inadequate. This algorithm aligns the texture stiffness force \mathbf{F}_M with the texture gradient in direction, and it makes algorithm A more vulnerable to the instability problem when applied to our height maps. By contrast, algorithm B renders a virtual surface with high stability; it adjusts only the magnitude of frictional force using the normalized texture height. However, this algorithm is less effective in conveying the fine roughness caused by small height variations on the surface.

Based on these observations, we propose a new force rendering algorithm that lies somewhere between algorithm A and B. Our approach modulates the magnitude of the frictional force as does algorithm B. The difference is that the magnitude depends on the gradient of height \mathbf{n}_t , not the normalized height, as follows:

$$\mathbf{F}_t = \mathbf{F}_M + \mathbf{F}_B + (1 - \cos \theta_t) \mathbf{F}_F, \quad (16)$$

where θ_t is the angle between the moving direction \mathbf{v}_T of the haptic interface point (HIP) and \mathbf{n}_t . If the normalized height has very small local fluctuation, the friction force variation rendered by algorithm B is imperceptible. It can be even so when the spatial frequency of the height map is sufficient to elicit a roughness sensation. By contrast, θ_t is related more to the spatial frequency of height, not the normalized height itself. This enables our algorithm to deliver a fine roughness sensation more faithfully than algorithm B.

B. STABILITY ANALYSIS

The three algorithms can be compared as to stability by examining the conservativity of the force field generated by each algorithm [5]. When an algorithm creates a non-conservative force field, the energy confined there might create artifacts. For this test, we calculate the energy gain following a cyclic trajectory shown in Figure 9. Algorithm A is tested using path A, and the two friction-based algorithms (B and the proposed algorithm) are tested on path B.

Both paths are determined to maximize the energy gain following a cyclic trajectory. In path A, we move from the origin in the tangential direction by the quarter wavelength of the sinusoidal height map. This maximizes the texture stiffness force \mathbf{F}_M rendered by Algorithm A, so maximizing the energy gain. For the friction-based algorithms, energy gain occurs only in the pre-sliding regime; the energy is dissipated during the sliding regime. Thus, we limit the tangential movement to d_x^{max} , the maximum displacement in the pre-sliding regime. The maximum vertical displacement d_z^{max} is limited by the thickness of the texture material.

For easy calculation, we use a sinusoidal height map $h(x) = A \sin(2\pi x/L)$. Then $|F_M|$ of algorithm A is

$$|F_m| = \begin{cases} \kappa A \sin\left(\frac{2\pi x}{L}\right) & \text{for } z < 0 \\ \kappa \left(A \sin\left(\frac{2\pi x}{L}\right) - z \right) & \text{for } 0 \leq z \leq h(x) \\ 0, & \text{for } z > h(x) \end{cases} \quad (17)$$

while κ is the stiffness coefficient of the material. Also \mathbf{n}_t is

$$\mathbf{n}_t = \frac{\left[\frac{-2\pi A}{L} \cos\left(\frac{2\pi x}{L}\right), 1 \right]^T}{\sqrt{1 + \left(\frac{2\pi A}{L} \cos\left(\frac{2\pi x}{L}\right) \right)^2}}. \quad (18)$$

Then the energy generated by algorithm A while cycling through path A is (see Appendix A for derivation)

$$\Delta E = \frac{\kappa A^2}{2} + \frac{\kappa L^2}{4\pi^2} \left(1 - \sqrt{\frac{L^2 + 4A^2\pi^2}{L^2}} \right). \quad (19)$$

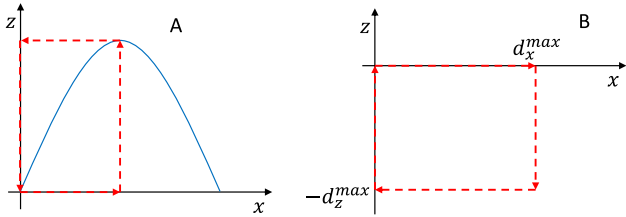


FIGURE 9. Paths used to calculate the energy gain of force-feedback texture rendering algorithms. Red dashed lines represent the cyclic trajectories for energy integration with direction, and the blue line in path A is the texture height map. 2D trajectories are chosen for easier calculation. d_x^{\max} is the tangential displacement during the stick phase, and d_z^{\max} is the maximum normal displacement of texture limited by the thickness of the texture material.

(19) proves that algorithm A generates a non-conservative force field³ following path A. If a haptic texture rendering system is generative, the accumulated internal energy might introduce undesirable buzzing and other kinds of noise during exploration on a textured surface [5]. These phenomena can degrade the perceived stability of the textured surface [46]. Depending on the values of κ , A , L and the type of haptic device, the energy gain in (19) can result in perceptible artifacts. Generally, algorithm A generates non-conservative force fields and is known to cause the feel of an active surface with other height maps [5].

For algorithm B and the proposed algorithm, only the friction force F_F generates energy following path B. We calculate the upper bound of the energy gain instead of the exact value since the Dahl model cannot be simplified to a closed form. The friction force F_F consists of the Dahl model term F_D and the viscous friction term $\mu_b v_T$. The upper bound of F_D is $F_C = \mu_k F_{norm}^{\max} = \mu_k \kappa d_z^{\max}$. The upper bound of $\mu_b v_T$ is $\mu_b v_{final}$, where v_{final} is the final velocity in the pre-sliding regime and d_z^{\max} is the maximum normal displacement limited by the thickness of the texture material. Then the upper bound of the energy gain is,

$$\Delta E \leq (\mu_b v_{final} + \mu_k \kappa d_z^{\max}) d_x^{\max} \quad (20)$$

for algorithm B and

$$\Delta E \leq 2 (\mu_b v_{final} + \mu_k \kappa d_z^{\max}) d_x^{\max} \quad (21)$$

for the proposed algorithm, where d_x^{\max} is the tangential displacement during the pre-sliding regime. Considering that d_x^{\max} is generally in the range from 10^{-2} m to 10^{-6} m and d_z^{\max} is limited by the thickness of the texture material, the energy gain is negligible. Thus, the proposed algorithm generates highly stable texture surfaces.

C. VIBRATION SYNTHESIS

To synthesize the contact acceleration of texture, we use the same filter coefficients \mathbf{h} optimized in Section II-A, but in the reverse direction (Figure 10) as described in [29]. This filter synthesizes the contact acceleration of the target material by

³In physics, we call a system conservative if and only if the system exerts zero work along any cycle.

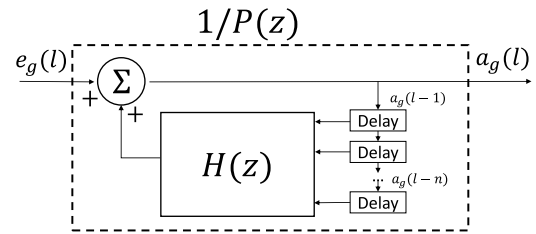


FIGURE 10. Diagram of vibration synthesis using an LPC-based model [29]. The inverse model $1/P(z)$ computes the contact vibration $a_g(l)$ of texture by adding white noise e_g to the filtered value of $a_g(l)$ using the filter $H(z)$.

adding the inner product between \mathbf{h} and the contact acceleration history $\mathbf{a}_g(l)$ with the white noise e_g . The power of the white noise is the same as that of the residual $e(k)$ resulted from the modeling process.

The filter coefficients for the current normal force F_{Norm} and the tangential velocity v_T are determined by barycentric interpolation using F_{Norm} and v_T as the interpolation variables. The tangential velocity can be acquired using the position encoders of the force-feedback device and the device kinematics. However, estimating the normal force applied by the user requires an additional force sensor. Instead, we use the normal force generated by the force-feedback algorithm. Given the values of F_{Norm} and v_T , we find four neighbor LPC filters and interpolate their coefficients individually to generate a new filter. Then, we synthesize the next contact acceleration sample using the new filter, the previous samples of contact acceleration, and white noise input. As long as the user maintains the contact and moves faster than a threshold velocity (1 cm/s), this process is repeated to synthesize a continuous vibration signal.

D. HYBRID RENDERING ALGORITHM

Computations for force feedback and vibration synthesis are performed by two dedicated but asynchronous threads because of their different requirements for update rate. In our current implementation on a regular PC (CPU: Intel I7 7700, RAM: 16 GB), the update rate for force feedback is approximately 2 kHz, and the sampling rate for vibration synthesis is 10 kHz. These high values are sufficient for haptic stimulus reconstruction without perceptual artifacts. In each iteration, the force feedback thread computes v_T and F_{Norm} . Based on their values, the force feedback thread turns the vibration synthesis thread on or off.

As mentioned earlier in Section II-A, the simultaneous rendering of force and vibration signals may exaggerate the texture sensation because they are modeled independently. To prevent this problem, we apply a low-pass filter with the cut-off frequency f_c to the force signal and ensure that force feedback delivers information only on the inhomogeneous components of the texture. The vibration signal is high-pass filtered also with f_c to preserve only the homogeneous characteristics. This process is for cross-talk removal in the viewpoint of the duplex theory of texture perception. After filtering, the signals are sent to their target devices.

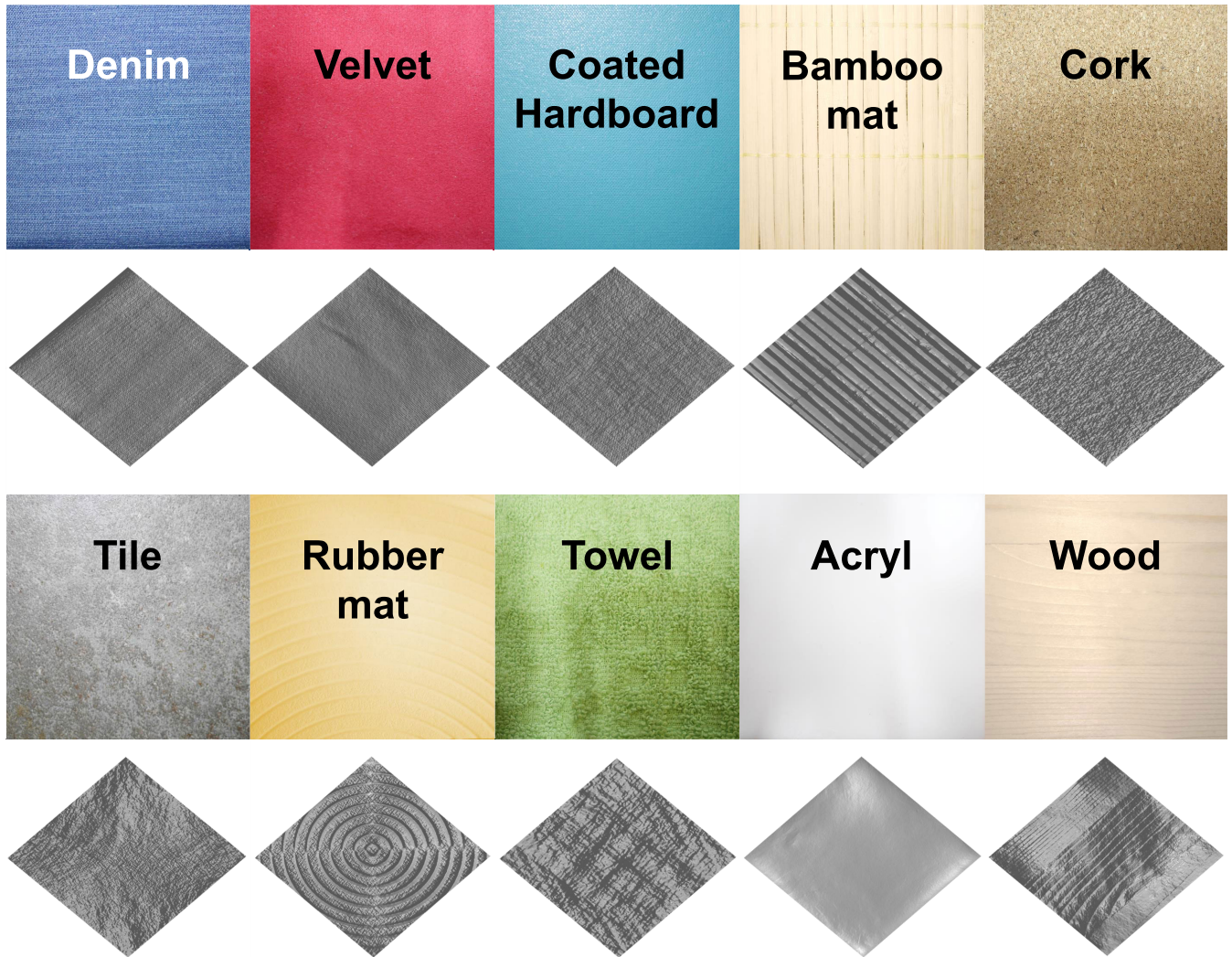


FIGURE 11. Texture materials used in user studies. The first and third rows show the photographs of the materials. The second and fourth rows show corresponding height maps captured by our photometric stereo system.

IV. USER STUDY 1: COMPARISON OF FORCE-FEEDBACK RENDERING ALGORITHMS

We compared three force-feedback texture rendering algorithms (Algorithm A: Norm, Algorithm B: Fric1, and the proposed algorithm: Fric2) described in Section III-A by means of a user study. The aim was finding the best match for the LPC-based vibration texture rendering algorithm used in our hybrid haptic texture framework. The emphasis was on the perceived realism of virtual textures, which was evaluated as the subjective similarity between virtual textures and their corresponding real textures. This study (and the next one) was approved by the institutional review board of the author's institution (PIRB-2019-E014).

A. METHODS

We used ten textured materials shown in Figure 11. They were selected for diversity. The ten materials can be classified

into three types of textiles (denim, velvet, towel), three types of wood (bamboo, cork, wood), one rubber (rubber mat), one plastic (acryl), one tile (tile), and one paper (coated hardboard). A half of them (bamboo, cork, rubber mat, tile, and wood) had fine visible geometric features while the others did not. Both stiffness and friction characteristics were also widely different. Based on the stiffness characteristics, the materials can be into three groups (Hard: acryl, bamboo mat, tile, wood, Medium: coated hardboard, denim, velvet, Soft: cork, rubber mat, towel). Similarly friction characteristics can also be divided into three groups (High: cork, rubber mat, tile, Medium: bamboo mat, denim, towel, wood, Low: acryl, coated hardboard, velvet).

We recruited 15 participants (9 males and 6 females; 19–31 years old with an average of 22.9; all right-handed) who had no experience in using haptic interfaces. None of them reported any existing sensory or motor impairment.

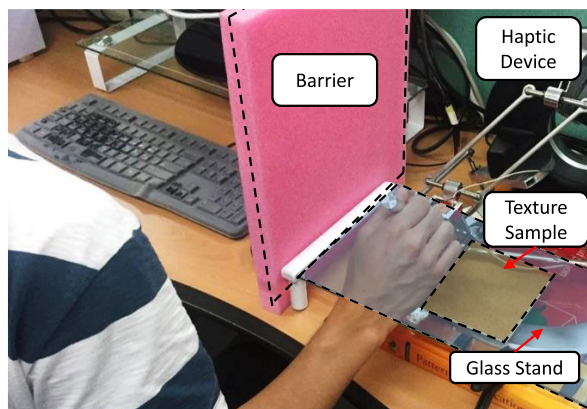


FIGURE 12. Experimental setup of the user studies.

The participants all signed an informed consent form after we explained the goals and procedure of the experiment. The participants were paid KRW 15,000 (approximately USD 13) after the experiment.

A stand was placed in front of the Omega.3 device to balance the height difference between real and virtual textured surfaces (Figure 12). Participants operated the Omega.3 device to stroke texture samples (both virtual and real) while holding its grip. Their view to the texture samples was blocked by a screen. Instead, the stylus position was displayed relative to the real and virtual textures on the monitor using a spherical cursor and two white rectangles. Participants wore noise-canceling headphones that played white noise to remove auditory cues.

Participants were asked to assess the subjective similarity between real and virtual textures under three criteria: roughness, friction, and overall similarity. Roughness and friction are two major dimensions of texture perception, and all force-feedback texture rendering algorithms control the roughness or friction of a virtual surface to elicit texture sensations. The meanings of the three criteria were explained to participants using real texture samples for clear understanding. Each measure was rated in a scale of 0–100 (0: completely different and 100: identical).

The experiment consisted of training and main sessions. In the training session, the ten virtual texture samples were rendered one by one in random order. Participants were asked to stroke each virtual texture while applying adequate pressure on the surface (0 N–12 N; maximum linear force of the Omega.3 device) for ten seconds. It was to prevent inexperienced participants from damaging the real texture samples and penetrating the virtual texture surfaces excessively. The main session contained 90 trials (10 materials \times 3 rendering methods \times 3 repetitions). The order of the trials was randomized for each participant. In each trial, participants evaluated the similarity between a virtual texture and its corresponding real texture. Their positions were shuffled randomly. Participants could freely explore the real and virtual textures. As discussed in Section III-B, Algorithm Norm has a high possibility of inducing unstable artifacts. If a participant

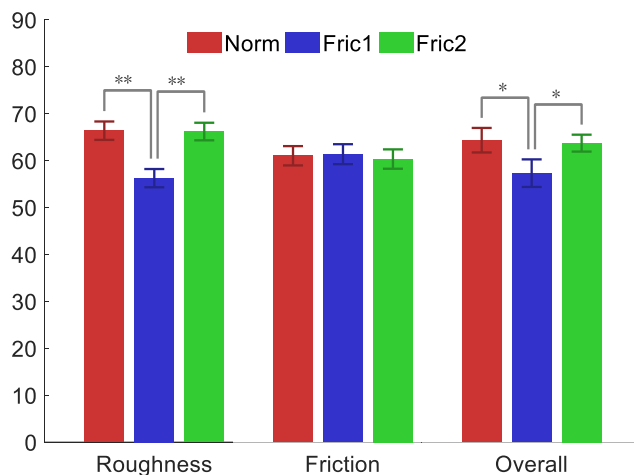


FIGURE 13. Average similarity scores of user study 1 for the three criteria. Error bars indicate 95% confidence intervals. Pairs grouped by asterisks were significantly different according to Tukey's HSD tests (* : $0.01 < p < 0.05$ and ** : $p < 0.01$).

experienced such an instability issue, we gave a one-minute break and resumed the trial.

Participants were allowed to take a break whenever necessary. On average, the experiment took approximately 80 minutes for each participant.

B. RESULTS

The average similarity scores for the three rendering methods are shown in Figure 13 for the three criteria. No significant outlier was found by a boxplot inspection. Error bars indicate 95% confidence intervals.

The average similarity scores for friction were all similar among the three rendering methods (Norm: 61.0, Fric1: 61.6, Fric2: 60.3). However, in roughness (Norm: 66.4, Fric1: 56.3, Fric2: 66.2) and overall similarity (Norm: 64.3, Fric1: 57.3, Fric2: 63.7), Fric1 showed a lower average score than the other two. For statistical analysis, we performed a two-way repeated-measure ANOVA on each measure using the texture material and the rendering method as independent variables. The data for all three measures passed the Shapiro-Wilk normality test and Mauchly's sphericity test, which are the prerequisites for ANOVA. Table 1 shows the results of ANOVA for each measure.

Regarding the similarity score for roughness, only the effect of the rendering method was found to be significant. The roughness similarity score depended mostly on the rendering method ($\eta^2 = 0.071$), rather than the texture material ($\eta^2 = 0.043$) or the interaction between them ($\eta^2 = 0.017$).

By contrast, the similarity score for friction exhibited significant correlation with the texture material and the interaction term, but not with the rendering method. In terms of effect sizes, the variation of the friction similarity score mostly came from the texture material ($\eta^2 = 0.154$). The rendering method showed almost no effect on the friction similarity score ($\eta^2 < 0.001$).

TABLE 1. ANOVA statistics of user study 1.

Roughness				
Factor	Df	<i>F</i>	<i>p</i>	η^2
Method*	2	26.873	<0.001	0.071
Material	9	1.433	0.181	0.043
Method:Material	18	1.340	0.163	0.017

Friction				
Factor	Df	<i>F</i>	<i>p</i>	η^2
Method	2	0.292	0.749	<0.001
Material*	9	3.977	<0.001	0.154
Method:Material*	18	3.429	<0.001	0.029

Overall				
Factor	Df	<i>F</i>	<i>p</i>	η^2
Method*	2	4.693	0.017	0.027
Material*	9	3.316	<0.001	0.086
Method:Material*	18	1.980	0.011	0.041

Significant factors are marked by *.

Both the two main factors and their interaction had significant effects on the overall similarity score. The material type had the largest effect size ($\eta^2 = 0.086$), while the interaction term and the rendering method had medium ($\eta^2 = 0.041$) and small effect size ($\eta^2 = 0.027$) respectively.

To direct the focus to the effect of the rendering method, we ran Tukey's HSD test for post-hoc multiple comparisons on the roughness and overall similarity score. In both measures, the differences between Norm and Fric2 were not significant, but Norm and Fric2 showed statistically higher scores than Fric1 (Figure 13).

As we also found the significant interaction effect on the friction and overall similarity scores, we analyzed the effect of the rendering method for every material using a simple effect test. For the friction similarity score, the rendering method was found to be significant only with Rubber mat ($p = 0.02$). The rendering method significantly affected the overall similarity scores for Rubber mat ($p < 0.001$), Tile ($p = 0.05$), and Towel ($p = 0.05$).

C. DISCUSSION

The rendering method showed a significant effect on the roughness similarity score. The Fric1 method scored much lower than the other two methods across all the texture samples. The Fric1 algorithm modulates the magnitude of friction force depending on the normalized texture height and can simulate a rough texture when the height variation is large compared to the texture thickness. However, if the height variation is small and changes quickly, the fluctuation in the friction force may be too small for perception. As a result, a user cannot feel the roughness adequately from the virtual surface. Actually, materials with visible geometry of height variations, such as Bamboo mat and Rubber mat (Figure 11), showed smaller score differences between Fric1 and Fric2 than the other materials.

The Fric2 algorithm also adjusts the magnitude of frictional force, but using the gradient of the height map. Consequently, Fric2 results in generally a greater gain than Fric1, while maintaining good passivity. Even when the height variation is small compared to the surface thickness, the Fric2 algorithm can deliver the rough feel with improved detail.

The results of user study 1 indicate that the Norm and Fric2 methods are more adequate for hybrid texture rendering with similar perceptual quality. Since Fric2 has better rendering stability than Norm, Fric2 must be our choice.

The friction similarity score was not significantly affected by the rendering method except for Rubber mat. Instead, the texture material had a significant impact on the score. For all the materials except Acryl, the higher the maximum friction force, the lower the similarity score.

One interesting finding is that the participants could not discern the frictional difference among the three rendering algorithms. Norm uses the friction force calculated from the Dahl model directly, but Fric1 and Fric2 modulate the magnitude of friction force for texture rendering. It appears that these high-frequency friction modulations did not contribute to the perception of different frictional properties but to the perception of different roughness properties [50], [51]. It is presumably due to our prior knowledge that friction information is contained in a relatively low-frequency band of surface response force.

The experimental results about the overall similarity score, including the scores and the ANOVA results, lie between the results of the roughness and the friction score. This tendency is natural as roughness and friction are the two major perceptual dimensions for haptic texture. Similarly to roughness, the rendering method was significant for overall similarity, but with a much smaller effect size.

The simple effect tests showed that the rendering method highly affected the three texture materials of Rubber mat, Tile, and Towel. These surfaces had both large grooves and small asperities on the surface and so greater height variations than the other materials.

V. USER STUDY 2: ASSESSING THE REALISM OF HYBRID HAPTIC TEXTURE

In user study 1, we compared the three force-feedback texture rendering algorithms for hybrid texture rendering. The results showed that the Fric2 algorithm is the most suitable. In user study 2, we evaluated the realism of our hybrid framework in comparison to the previous haptic texture rendering algorithms that use either force or vibration feedback.

A. METHODS

User study 2 had the same experiment apparatus and procedure as user study 1. We recruited 20 participants (11 males and 9 females; 20–27 years old with an average age of 23.6 years; all right-handed) who were all inexperienced in using haptic interfaces.

For hybrid haptic texture rendering (Hybrid), we used the Fric2 algorithm. To render force-only virtual textures (FF),

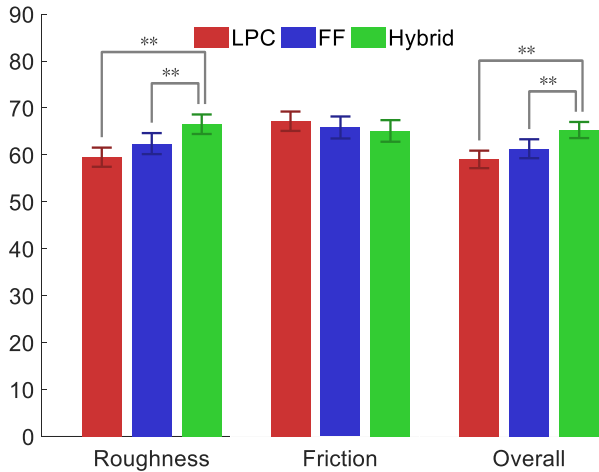


FIGURE 14. Average similarity scores obtained in user study 2 for three comparison criteria. Error bars indicate 95% confidence intervals. Pairs grouped by asterisks were significantly different by Tukey's HSD tests (* : $0.01 < p < 0.05$, ** : $p < 0.01$).

we rendered the force signal generated by the force generation module described in Section III-A with no hybrid filtering or vibration feedback. For vibration-only textures (LPC), we applied no texture rendering algorithm in the force feedback module to render only the flat surface. No hybrid filtering was applied either. The physical parameters for the Dahl model and the Hunt-Crossley model were same for the three rendering methods.

The same training session as in user study 1 followed the main session consisting of 90 trials (10 materials \times 3 rendering methods \times 3 repetitions).

B. RESULTS

The average similarity scores for the three rendering methods are shown for each criterion in Figure 14. No significant outliers were found by boxplot inspection. We applied two-way repeated-measure ANOVA on the data of each measure using the rendering method and the texture material as independent variables. The data for all three measures passed the Shapiro-Wilk normality test and Mauchly's sphericity test. ANOVA results for each measure are detailed in Table 2.

The average scores for roughness similarity were ordered as Hybrid (66.5) $>$ FF (62.4) $>$ LPC (59.5), and this was statistically significant. Tukey's HSD test for post-hoc multiple comparisons showed that Hybrid had a significantly better score than FF and LPC (Figure 14). The texture material was also significant for roughness similarity, as well as the interaction term. The effect sizes in terms of η^2 showed that the interaction term was the most dominant factor (Table 2).

For friction similarity, the rendering method did not cause significant differences in the average scores. The texture material and the interaction term were significant, and the effect size of the texture material was almost double of the effect size of the interaction term.

The overall similarity scores showed a very similar trend to that for roughness similarity. The effect of the rendering

TABLE 2. ANOVA statistics of user study 2.

Roughness				
Factor	Df	F	p	η^2
Method*	2	13.121	<0.001	0.021
Material*	9	2.279	0.021	0.048
Method:Material*	18	4.581	<0.001	0.055

Friction				
Factor	Df	F	p	η^2
Method	2	1.685	0.204	0.002
Material*	9	6.324	<0.001	0.118
Method:Material*	18	8.061	<0.001	0.070

Overall				
Factor	Df	F	p	η^2
Method*	2	13.182	<0.001	0.022
Material	9	1.625	0.115	0.036
Method:Material*	18	5.066	<0.001	0.073

Significant factors are marked by *.

method was statistically significant, and the average scores were ordered as Hybrid (65.30) $>$ FF (61.3) $>$ LPC (59.0). Hybrid had a significantly better score than FF and LPC according to Tukey's tests (Figure 14). The texture material, however, was not significant for the overall similarity. The effect size was greatest with the interaction term (Table 2).

In addition, we examined the interaction effect on all three similarity scores by performing a simple effect test on the rendering method for each material. For roughness similarity, Acryl ($p = 0.04$), Bamboo ($p < 0.01$), Velvet ($p < 0.01$), and Wood ($p < 0.01$) were significantly affected by the rendering method. Furthermore, for Acryl, Bamboo, and Wood, both FF and Hybrid outperformed LPC by pairwise comparisons. For Velvet, LPC and Hybrid were significantly better than FF. In case of friction similarity, the rendering method was significant for Tile ($p < 0.01$), Towel ($p = 0.01$), and Velvet ($p < 0.01$). For Tile, both LPC and Hybrid were significantly more effective than FF. For Towel, FF and Hybrid were superior to LPC. For Velvet, the friction similarity scores were ordered as LPC $>$ FF $>$ Hybrid with significance. Finally, for overall similarity, Bamboo ($p < 0.01$), Velvet ($p < 0.01$), and Wood ($p < 0.01$) had the significant effect of the rendering method. For Bamboo, Hybrid gave a significantly higher score than FF and LPC. For Velvet, the scores were significantly higher with Hybrid and LPC than FF. For Wood, FF and Hybrid resulted in higher scores than LPC.

C. DISCUSSION

For the texture rendering performance of roughness, the Hybrid method showed the best subjective performance with statistical significance, and the other two methods, LPC and FF, were not distinguished. We can obtain more insights by examining the scores for individual texture materials shown in Table 3, where the materials are grouped into six homogeneous and four inhomogeneous ones. For all the

TABLE 3. Similarity scores about roughness for each texture.

Inhomogeneous						
Method	Bamboo	Denim	Rubbermat	Tile	Towel	Wood
LPC	44.00	58.67	58.22	61.33	55.78	56.67
FF	60.00	69.22	70.56	54.64	62.67	69.67
Hybrid	62.11	70.78	73.22	68.78	65.22	71.89

Homogeneous				
Method	Acryl	Hardboard	Cork	Velvet
LPC	66.44	61.87	65.11	67.00
FF	72.89	57.22	57.16	50.02
Hybrid	72.78	58.24	61.33	60.98

TABLE 4. Overall similarity scores for each texture.

Inhomogeneous						
Method	Bamboo	Denim	Rubbermat	Tile	Towel	Wood
LPC	52.49	58.87	58.44	59.20	53.76	52.02
FF	62.69	70.62	67.18	55.20	60.60	68.22
Hybrid	70.18	65.00	69.62	66.47	57.84	70.09

Homogeneous				
Method	Acryl	Hardboard	Cork	Velvet
LPC	65.09	57.29	60.18	73.04
FF	69.93	51.91	55.11	51.64
Hybrid	64.96	57.84	65.04	65.91

inhomogeneous materials, FF received a higher score than LPC except Tile, which generates very strong contact vibration for texture. Conversely, LPC received higher scores than FF for all the homogeneous textures except Acryl. Hence, we can deduce that the FF algorithm is appropriate for inhomogeneous textures and LPC is suitable for homogeneous ones; also see [52] for the detailed and categorized comparisons between force and vibration texture rendering as to perceived realism. The results also indicate that our hybrid rendering method appropriately inherits the respective advantages of the two methods, leading to the superior performance for rendering texture roughness.

For friction similarity, the three texture rendering methods resulted in similar perceptual performance without significant effects, as user study 1.

For overall similarity, the Hybrid method outperformed FF and LPC with statistical significance. These results are well aligned with the results of roughness and friction similarity. Table 4 displays the average overall similarity scores for each material and rendering method. While the overall similarity scores of LPC and FF largely depend on the texture homogeneity, the Hybrid method shows consistently higher or comparable scores.

The simple effect test on the interaction effects showed that in most materials except Velvet about friction score, Hybrid method got higher or comparable scores to the other rendering methods. Thus, the previous discussion on the main effects holds for the interaction effects. FF method was more adequate for rendering inhomogeneous materials such as Bamboo, Wood, Towel, while LPC resulted in higher

similarity score for Velvet, which can be categorized as homogeneous texture.

The above results allow us to conclude that our hybrid rendering method is more effective than the prior ones in terms of realism (similarity to real textures) and the scope of application (the type of materials).

When users are asked to compare the haptic similarity between two identical real objects or textures, they generally give around 70—80 out of 100 because of the general regression bias. For very different real textures, the scores lie in the range of 20—30 out of 100 [12], [35], [41]. In user study 2, the grand mean of the overall similarity scores was 65.30 for our hybrid rendering, with the highest of 70.2 (bamboo) and the lowest of 57.8 (towel and hardboard). This indicates that we still have room for further improvement in terms of realism for haptic texture modeling and rendering.

VI. CONCLUSION

In this paper, we have proposed a hybrid framework for haptic texture modeling and rendering using simultaneous force and vibration feedback. This framework is suitable for the effective and realistic modeling and rendering of both homogeneous and inhomogeneous textures. A geometry-based texture modeling algorithm and a new force-feedback texture rendering algorithm using friction modulation are presented, as well as parameter identification procedures for the Dahl friction model and the Hunt-Crossley stiffness model. Moreover, a hybrid texture rendering algorithm is designed and validated, which blends force and vibration feedback while removing the perceptual overlap between the two modalities.

Our methods and algorithms have been evaluated by two user studies as to the perceived similarity between real and virtual textures. User study 1 compared three force-feedback texture rendering algorithms and indicated that the proposed force-feedback rendering algorithm, which also preserves fine inhomogeneous textural features, is more suitable for our hybrid texture framework. In user study 2, we compared the textures rendered using the hybrid framework with the textures rendered using either force or vibration, which represents the previous practice. The results demonstrated that the hybrid texture framework achieves higher realism and a wider application scope.

Our hybrid framework still has room for further improvement. First, the hybrid texture rendering algorithm is based on the simplified physics, and it may be replaced with a more advanced one. Data-driven techniques using machine learning can be a good solution for that. Second, a thermal modeling and rendering model suitable for haptic texture has been left as future work. Adding thermal feedback may greatly enhance the realism of virtual textures.

APPENDIX A

DERIVATION OF THE ENERGY GAIN ON PATH A

The energy gain by algorithm A while cycling through path A can be calculated by integrating the inner product between the material's stiffness force \mathbf{F}_M and the displacement following

the given path, such that.

$$\begin{aligned}\Delta E &= \oint_{\text{pathA}} \mathbf{F}_M \cdot ds \\ &= \oint_{\text{pathA}} |F_M| \mathbf{n}_t \cdot ds,\end{aligned}\quad (22)$$

where ds is infinitesimal arc length on path A.

The line integral of (22) consists of four integrals,

$$\begin{aligned}\oint_{\text{pathA}} |F_M| \mathbf{n}_t \cdot ds \\ &= \int_A^0 |F_M| \mathbf{n}_t \cdot dz \Big|_{x=0} + \int_0^{\frac{L}{4}} |F_M| \mathbf{n}_t \cdot dx \Big|_{z=0} \\ &\quad + \int_0^A |F_M| \mathbf{n}_t \cdot dz \Big|_{x=\frac{L}{4}} + \int_{\frac{L}{4}}^0 |F_M| \mathbf{n}_t \cdot dx \Big|_{z=A}.\end{aligned}\quad (23)$$

As the HIP does not penetrate the material when $x = 0$ and $z = A$, $|F_M|$ for the first and the last integral is zero. Thus, only the second and third integral contribute to the line integral.

$$\begin{aligned}\oint_{\text{pathA}} |F_M| \mathbf{n}_t \cdot ds \\ &= \int_0^{\frac{L}{4}} |F_M| \mathbf{n}_t \cdot dx \Big|_{z=0} + \int_0^A |F_M| \mathbf{n}_t \cdot dz \Big|_{x=\frac{L}{4}}.\end{aligned}\quad (24)$$

By substituting (17) and (18) into the first term on the right side of (24),

$$\begin{aligned}\int_0^{\frac{L}{4}} |F_M| \mathbf{n}_t \cdot dx \Big|_{z=0} \\ &= \int_0^{\frac{L}{4}} \frac{-2\pi A^2 \kappa \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi x}{L}\right)}{\sqrt{L^2 + 4\pi^2 A^2 \cos^2\left(\frac{2\pi x}{L}\right)}} dx.\end{aligned}\quad (25)$$

Factoring out the constants gives

$$= -\frac{2\pi A^2 \kappa}{L} \int_0^{\frac{L}{4}} \frac{\sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi x}{L}\right)}{\sqrt{1 + \frac{4\pi^2 A^2 \cos^2\left(\frac{2\pi x}{L}\right)}{L^2}}} dx.\quad (26)$$

For the integrand $\frac{\sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi x}{L}\right)}{\sqrt{1 + \frac{4\pi^2 A^2 \cos^2\left(\frac{2\pi x}{L}\right)}{L^2}}}$, use $u = \frac{2\pi x}{L}$ and $du =$

$$\frac{2\pi}{L} dx \\ = -A^2 \kappa \int_0^{\frac{\pi}{2}} \frac{\sin(u) \cos(u)}{\sqrt{1 + \frac{4\pi^2 A^2 \cos^2(u)}{L^2}}} du.\quad (27)$$

For the integrand $\frac{\sin(u) \cos(u)}{\sqrt{1 + \frac{4\pi^2 A^2 \cos^2(u)}{L^2}}}$, substitute $s = \frac{4\pi^2 A^2 \cos^2(u)}{L^2} + 1$ and $ds = -\frac{8\pi^2 A^2 \sin(u) \cos(u)}{L^2} du$

$$= \frac{\kappa L^2}{8\pi^2} \int_{1+\frac{4\pi^2 A^2}{L^2}}^1 \frac{1}{\sqrt{s}} ds.\quad (28)$$

Then the value of (27) is

$$\frac{\kappa L^2}{4\pi^2} \sqrt{s} \Big|_{1+\frac{4\pi^2 A^2}{L^2}}^1 = \frac{\kappa L^2}{4\pi^2} \left(1 - \sqrt{1 + \frac{4\pi^2 A^2}{L^2}}\right).\quad (29)$$

The remaining second term on the right side of (23) is

$$\begin{aligned}\int_0^A |F_M| \mathbf{n}_t \cdot dz \Big|_{x=\frac{L}{4}} &= \int_0^A \kappa \left(A \sin\left(\frac{2\pi x}{L}\right) - z\right) \\ &= \frac{\kappa A^2}{2}.\end{aligned}\quad (30)$$

Summing up (29) and (30) gives the total energy gain of algorithm A on path A.

$$\Delta E = \frac{\kappa L^2}{4\pi^2} \left(1 - \sqrt{1 + \frac{4\pi^2 A^2}{L^2}}\right) + \frac{\kappa A^2}{2}.\quad (31)$$

REFERENCES

- [1] M. D. R. Minsky, *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display*. Cambridge, MA, USA: MIT Press, 1995.
- [2] A. Hardwick, S. Furner, and J. Rush, "Tactile display of virtual reality from the world wide Web—A potential access method for blind people," *Displays*, vol. 18, no. 3, pp. 153–161, May 1998.
- [3] T. H. Massie, "Initial haptic explorations with the phantom: Virtual touch through point interaction," Ph.D. dissertation, Dept. Mech. Eng., MIT, Cambridge, MA, USA, 1996.
- [4] V. Hayward and D. Yi, "Change of height: An approach to the haptic display of shape and texture without surface normal," *Experim. Robot.*, vol. 54, pp. 570–579, Jun. 2003.
- [5] G. Campion, *The Synthesis of Three Dimensional Haptic Textures: Geometry, Control, and Psychophysics*. London, U.K.: Springer, 2011, pp. 73–97.
- [6] J. Siira and D. K. Pai, "Haptic texturing—A stochastic approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Minneapolis, MN, USA, Oct. 1996, pp. 557–562.
- [7] J. P. Fritz and K. E. Barner, "Stochastic models for haptic texture," in *Proc. Telemanipulator Telepresence Technol.*, Boston, MA, USA, Dec. 1996, pp. 34–44.
- [8] A. Crossan, J. Williamson, and R. Murray-Smith, "Haptic granular synthesis: Targeting, visualisation and texturing," in *Proc. 8th Int. Conf. Inf. Visualisation*, London, U.K., 2004, pp. 527–532.
- [9] A. M. Okamura, J. T. Dennerlein, and R. D. Howe, "Vibration feedback models for virtual environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, BE, USA, 1998, pp. 674–679.
- [10] J. Lang and S. Andrews, "Measurement-based modeling of contact forces and textures for haptic rendering," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 3, pp. 380–391, Mar. 2011.
- [11] V. L. Guruswamy, J. Lang, and W.-S. Lee, "IIR filter models of haptic vibration textures," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 1, pp. 93–103, Jan. 2011.
- [12] J. M. Romano and K. J. Kuchenbecker, "Creating realistic virtual textures from contact acceleration data," *IEEE Trans. Haptics*, vol. 5, no. 2, pp. 109–119, Apr. 2012.
- [13] A. Abdulali and S. Jeon, "Data-driven modeling of anisotropic haptic textures: Data segmentation and interpolation," in *Proc. EuroHaptics*, London, U.K., 2016, pp. 228–239.
- [14] R. H. Osgouei, S. Shin, J. R. Kim, and S. Choi, "An inverse neural network model for data-driven texture rendering on electrovibration display," in *Proc. IEEE Haptics Symp. (HAPTICS)*, San Francisco, CA, USA, Mar. 2018, pp. 270–277.
- [15] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures," *IEEE Trans. Haptics*, vol. 6, no. 1, pp. 81–93, Jan. 2013.
- [16] 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. IEEE Haptics Symp.*, Vancouver, BC, CA, Mar. 2012, pp. 385–391.

- [17] H. Culbertson, J. Unwin, B. E. Goodman, and K. J. Kuchenbecker, "Generating haptic texture models from unconstrained tool-surface interactions," in *Proc. World Haptics Conf. (WHC)*, Daejeon, KR, USA, Apr. 2013, pp. 295–300.
- [18] M. Hollins and S. R. Risner, "Evidence for the duplex theory of tactile texture perception," *Perception Psychophys.*, vol. 62, no. 4, pp. 695–705, Jan. 2000.
- [19] C. Basdogan, C. Ho, and M. A. Srinivasan, "A raybased haptic rendering technique for displaying shape and texture of 3D objects in virtual environments," in *Proc. ASME Dyn. Syst. Control Division*, vol. 61, Dallas, TX, USA, 1997, pp. 77–84.
- [20] R. L. Klatzky, S. J. Lederman, C. Hamilton, M. Grindley, and R. H. Swendsen, "Feeling textures through a probe: Effects of probe and surface geometry and exploratory factors," *Perception Psychophys.*, vol. 65, no. 4, pp. 613–631, May 2003.
- [21] S. Choi and H. Z. Tan, "Toward realistic haptic rendering of surface textures," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 40–47, Mar. 2004.
- [22] M. A. Otaduy and M. C. Lin, "A perceptually-inspired force model for haptic texture rendering," in *Proc. 1st Symp. Appl. Perception Graph. Visualizat.*, Los Angeles, CA, USA, 2004, pp. 123–126.
- [23] M. A. Costa, M. R. Cutkosky, and S. Lau, "Roughness perception of haptically displayed fractal surfaces," in *Proc. ASME Dyn. Syst. Control Division*, Orlando, FL, USA, vol. 69, no. 2, 2000, pp. 1073–1079.
- [24] S. S. Wall and W. S. Harwin, "Modelling of surface identifying characteristics using Fourier series," in *Proc. ASME Dyn. Syst. Control Division*, Nashville, TN, USA, 1999, pp. 65–71.
- [25] D. K. Pai and P. Rizun, "The WHaT: A wireless haptic texture sensor," in *Proc. 11th Symp. Haptic Interface Virtual Environ. Teleoperator Syst.*, Los Angeles, CA, USA, Oct. 2003, pp. 3–9.
- [26] Y. Ikei, K. Wakamatsu, and S. Fukuda, "Vibratory tactile display of image-based textures," *IEEE Comput. Graph. Appl.*, vol. 17, no. 6, pp. 53–61, Nov. 1997.
- [27] S. Andrews and J. Lang, "Interactive scanning of haptic textures and surface compliance," in *Proc. 6th Int. Conf. 3-D Digit. Imag. Modeling*, Montreal, QC, Canada, Aug. 2007, pp. 99–106.
- [28] S. Andrews and J. Lang, "Haptic texturing based on real-world samples," in *Proc. IEEE Int. Workshop Haptic, Audio Vis. Environ. Games*, Ottawa, ON, Canada, Oct. 2007, pp. 142–147.
- [29] J. M. Romano, T. Yoshioka, and K. J. Kuchenbecker, "Automatic filter design for synthesis of haptic textures from recorded acceleration data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA May 2010, pp. 1815–1821.
- [30] 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. IEEE Haptics Symp. (HAPTICS)*, Houston, TX, USA, Feb. 2014, pp. 319–325.
- [31] M. Hollins, S. Bensmaïa, and R. Risner, "The duplex theory of tactile texture perception," in *Proc. Fechner Day*, 1998, pp. 115–121.
- [32] G. Elliott, T. J. Rothenberg, and J. H. Stock, "Efficient tests for an autoregressive unit root," *Nat. Bureau Econ. Res.*, Cambridge, MA, USA, Tech. Working Paper 130, 1992.
- [33] N. Wiener, N. Wiener, C. Mathematician, N. Wiener, N. Wiener, and C. Mathématicien, *Extrapolation, Interpolation, Smoothing Stationary Time Series: With Engineering Applications*. Cambridge, MA, USA: MIT Press, 1949.
- [34] W. F. Trench, "An algorithm for the inversion of finite toeplitz matrices," *J. Soc. for Ind. Appl. Math.*, vol. 12, no. 3, pp. 515–522, Sep. 1964.
- [35] S. Shin and S. Choi, "Geometry-based haptic texture modeling and rendering using photometric stereo," in *Proc. IEEE Haptics Symp.*, San Francisco, CA, USA, Mar. 2018, pp. 262–269.
- [36] A. D. Spence and M. J. Chantler, "Optimal illumination for three-image photometric stereo using sensitivity analysis," *IEE Proc. Vis., Image Signal Process.*, vol. 153, no. 2, pp. 149–159, Apr. 2006.
- [37] J. A. Paterson, D. Claus, and A. W. Fitzgibbon, "BRDF and geometry capture from extended inhomogeneous samples using flash photography," in *Computer Graphics Forum*, vol. 24, Hoboken, NJ, USA: Wiley, 2005, pp. 383–391.
- [38] L. Zhang, Curless, Hertzmann, and Seitz, "Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multiview stereo," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, 2003, pp. 618–625.
- [39] H. Culbertson and K. J. Kuchenbecker, "Importance of matching physical friction, hardness, and texture in creating realistic haptic virtual surfaces," *IEEE Trans. Haptics*, vol. 10, no. 1, pp. 63–74, Aug. 2016.
- [40] S. Jeon, S. Choi, and M. Harders, "Rendering virtual tumors in real tissue mock-ups using haptic augmented reality," *IEEE Trans. Haptics*, vol. 5, no. 1, pp. 77–84, Jan. 2012.
- [41] S. Jeon, "Real stiffness augmentation for haptic augmented reality," *Presence, Teleoperators Virtual Environ.*, vol. 20, no. 4, pp. 337–370, Aug. 2011.
- [42] A. Haddadi and K. Hashtrudi-Zaad, "A new method for online parameter estimation of hunt-crossley environment dynamic models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, France, Sep. 2008, pp. 981–986.
- [43] S. Choi, L. Walker, H. Z. Tan, S. Crittenden, and R. Reifenberger, "Force constancy and its effect on haptic perception of virtual surfaces," *ACM Trans. Appl. Perception*, vol. 2, no. 2, pp. 89–105, Apr. 2005.
- [44] M. Mahvash and A. M. Okamura, "Friction compensation for a force-feedback telerobotic system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, 2006, pp. 3268–3273.
- [45] F. Wang, T. Hurst, D. Abramovitch, and G. Franklin, "Disk drive pivot nonlinearity modeling. II. Time domain," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, 1994, pp. 2604–2607.
- [46] S. Choi and H. Z. Tan, "Perceived instability of virtual haptic Texture. I. Experimental studies," *Presence, Teleoperators Virtual Environ.*, vol. 13, no. 4, pp. 395–415, Aug. 2004.
- [47] S. Choi and H. Z. Tan, "Perceived instability of virtual haptic Texture. II. Effect of collision-detection algorithm," *Presence, Teleoperators Virtual Environ.*, vol. 14, no. 4, pp. 463–481, Aug. 2005.
- [48] S. Choi and H. Z. Tan, "Perceived instability of virtual haptic texture: III. Effect of update rate," *Presence, Teleoperators Virtual Environ.*, vol. 16, no. 3, pp. 263–278, Jun. 2007.
- [49] C.-H. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient point-based rendering techniques for haptic display of virtual objects," *Presence, Teleoperators Virtual Environ.*, vol. 8, no. 5, pp. 477–491, Oct. 1999.
- [50] M. Hollins, A. Fox, and C. Bishop, "Imposed vibration influences perceived tactile smoothness," *Perception*, vol. 29, no. 12, pp. 1455–1465, Dec. 2000.
- [51] S. Asano, S. Okamoto, and Y. Yamada, "Vibrotactile stimulation to increase and decrease texture roughness," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 3, pp. 393–398, Jun. 2015.
- [52] S. Shin and S. Choi, "Effects of haptic texture rendering modalities on realism," in *Proc. 24th ACM Symp. Virtual Reality Softw. Technol.*, Tokyo, Japan, Nov. 2018, Art. no. 42.



SUNGHWAN SHIN (Student Member, IEEE) received the B.S. and Ph.D. degrees in computer science and engineering from the Pohang University of Science and Technology, in 2010 and 2019, respectively. He is currently working with Samsung Research. His research interests include modeling and rendering of haptic texture, multi-modal haptic rendering, and haptic rendering algorithm.



SEUNGMOON CHOI (Senior Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, in 1995 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, in 2003. He is currently a Professor of computer science and engineering with the Pohang University of Science and Technology (POSTECH). His general research interests include on haptic rendering and perception, with application to robotics, human-computer interaction, virtual reality, and consumer products. He received a 2011 Early Career Award from the IEEE Technical Committee on Haptics. He is an Associate Editor-in-Chief of the IEEE TRANSACTIONS ON HAPTICS, since 2020. He was the General Co-Chair of the IEEE Haptics Symposium, in 2014 and 2016.

•••