Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

Generating haptic texture using solid noise☆

Osama Halabi [a,*], Gulrukh Khattak [b]

[a] Department of Computer Science and Engineering, Qatar University, P.O.Box 2713, Doha, Qatar
[b] Electrical Engineering Department and Networks Research, University of Engineering and Technology Peshawar, P.O. Box 25120, Peshawar, Pakistan

ARTICLE INFO

Keywords:
Haptic texture
Noise-based texture
Solid noise
Texture simulation

ABSTRACT

Texture enhances haptic interaction by providing unique, distinguishable, and versatile surfaces. In computer haptics, texture can render environments more realistic and provide useful information. In this paper, an algorithm is proposed for virtual texture simulation by using solid noise, where only a few parameters need to be altered to generate a range of realistic and diverse textures by reproducing different frequencies similar to that of real vibrational signals in a virtual environment. The proposed method can capture the textural effect in a haptic simulation while retaining a simple overall geometry and stable update rate. This method also allows the user to change the texture at runtime and can be easily incorporated into any existing code and used in any traditional haptic device without affecting overall haptic-rendering performance. Moreover, the solid noise texture is independent of object geometry and can be applied to any shape without additional computations. We conducted a human-subject study to evaluate the recognition accuracy for each generated haptic texture as well as its realism and correspondence to real texture. The results indicated the high performance of the method and its ability to generate haptic textures with a very high recognition rate that were highly realistic.

1. Introduction

Texture makes an important contribution to the identification of the material. A massive wealth of textures can be found in the world. Hence, the texture is necessary for a virtual world designed to appear realistic. Texture can convey useful information and assist in perception. Touch is an important means of exploring the world. Haptic texture by itself does not reveal the complex sensations experienced due to reception mechanisms embedded in the skin. The dragging of pencil on paper or vibrations in a piece of fabric are felt by the subject in the texture of the material due to microscopic surface irregularities [1].

Past research has highlighted the importance of vibrotactile signals during haptic interaction. In writing with a pencil on grained paper, the impression of the textured surface arises out of the vibrations transmitted to the fingers [2]. Therefore, the vibrations experienced on the fingertips convey important information concerning this interaction, even though the finger does not directly touch the surface. As Katz noted, people have rich impressions of the surface but not the vibrations themselves [3].

-To create the perceptual illusion of touching real surfaces during virtual interaction, we need to generate appropriate vibrations. Many attempts have been made to recreate these interactions, but they either still lack the richness of real haptic feedback experienced during interactions with the physical world, or are too complex to implement and use in realtime.

Height maps have been used to generate textures over surfaces, but textures in such cases are predefined and cannot be changed at runtime [4]. Procedural methods can generate textures created at runtime [5,6]. Most past efforts that used procedural approaches succeeded in generating mutually differentiable textures with varying degrees of roughness. Most recent work on texture rendering has employed the approach of recording real textures and reproducing them using force, displacement, or acceleration data [3,7,8,9,10,11,12,13]. However, these proposals are limited to surfaces that have been previously measured, and do not work if we wish to simulate new surfaces known only through geometric and material specification [14]. Moreover, they require sensitive force-measuring instruments. Haptic texture has also been implemented by using shaders [6].

This paper proposes a method to help generate textures based on a simple solid noise function. The focus is on reproducing the frequency of real vibrational signals in a virtual environment. The proposed method can capture the textural effect in a haptic simulation while retaining a

☆ This paper was recommended for publication by Prof. G. Guangtao Zhai.
* Corresponding author.
E-mail address: ohalabi@qu.edu.qa (O. Halabi).

https://doi.org/10.1016/j.displa.2021.102048
Received 17 February 2021; Received in revised form 24 June 2021; Accepted 28 June 2021
Available online 3 July 2021

simple overall geometry and a stable update rate. The algorithm can be easily incorporated into any existing code and used in any traditional haptic device without affecting overall haptic-rendering performance. At the same time, it provides a set of parameters that can be used to create a vast array of textures.

The method proposed here is a simple approach that allows a user to change the texture at runtime and can be used in any existing code since solid noise texture is independent of object geometry and can be applied to any shape without additional computations. The important characteristics of any visual texture can be used to define appropriate parameter values for haptic texture simulation. Any number of textures can be included in a haptic scene, as the texture algorithm only needs to be executed for the point of contact. The method was tested using a set of experiments involving volunteers. The experiments verified that the proposed method can simulate the effect of haptic texture that can be felt and recognized by any user.

This paper is organized as follows. Section 2 provides essential background about haptic texture and related research. Section 3 discusses past research that is relevant to the current study. Section 4 explains in detail the proposed method for generating haptic texture using Perlin noise. Section 5 describes the experimental setup and procedure used for evaluation, and Section 6 presents its result. Section 7 provides a thorough discussion of the results, and Sections 8 and 9 conclude the paper with concluding remarks and future work, respectively.

2. Background

Computer haptics is the tangible counterpart of computer graphics. The underlying principles are similar: haptic rendering uses force where graphic rendering uses color, and haptic rendering uses a force feedback device whereas graphic rendering uses a graphic display. Many texture-generating techniques have been inspired by analogous techniques in modern computer graphics. In computer graphics, texture mapping adds realism to scenes by projecting a bitmap image onto the surfaces being rendered. The same approach can be applied to haptic textures that are generated. The sandpaper system developed by Minsky et al. was the first for haptic texture mapping in two dimensions [15]; Ruspini et al. later extended this to 3D scenes [16]. Using mathematical functions is another approach to create synthetic patterns. Ho et al. [17] and Costa et al. [18] explored the use of fractals to model natural textures.

In haptic interaction, a virtual scene is represented by translating physical features into variations in force. The appropriate forces to be returned are computed by simulation algorithms whenever an interaction occurs, and are applied through a haptic device. Thus, it is possible not only to convey the geometry of an object, but also its physical properties like stiffness, friction, mass, density, viscosity, and deformability. One of the intrinsic properties of any material is its texture.

2.1. Psychophysics of haptic texture

Texture can be defined as the feel of a material. It results from tiny irregularities on a surface that are characteristic of a material but are not a part of its geometric shape. As the finger scans a textured surface, vibrations are produced in the skin. These vibrations are then transduced by rapidly adapting mechanoreceptors in the skin. There are two types, the Meissner's corpuscles are the most common mechanoreceptors and their afferent fibers account for about 40% of the sensory innervation of the human hand. These corpuscles are particularly efficient in detecting low-frequency vibration (30–50 Hz). The Pacinian corpuscles on the other hand adopt more rapidly and allow response to high frequencies (250–350 Hz). As a result, it is on the basis of this vibrotactile signals that these textures are perceived. But to enable a haptic device to do so, a texture must first be converted into a mathematical equation to produce relevant frequencies.

A number of psychophysical studies [3] were carried out to determine this mathematical form. The role of vibratory texture coding differs when we explore texture using bare skin rather than a probe. With a probe in contact with the given surface, the resulting spatial map reflects the contours of the probe instead of those of the surface. The surface properties that constitute texture give rise to vibrations that are transmitted to the skin along with a rigid link. It has been observed that during the exploration of a surface by a bare finger, sensory cells on the skin perceive texture by decoding the spatial layout of textural irregularities in the form of a magnitude variation called the spatial-intensive method. On the contrary, while exploring a surface with a tool, textural perception is based on a vibratory code [2]. A haptic device uses a tool to explore a virtual environment; therefore, if touch means force in haptics, texture represents the vibrations of that force.

Haptic texture perception indicates that temporal variations in mechanical signals (vibrations) and spatially distributed stimulation of the skin contribute to perception [19,20,21,22]. Hollins et. Al. mentioned that vibrational and spatial stimulation both contribute to texture perception in what is called the duplex theory of tactile. Mechanisms for responding to the spatial component appear to be engaged at texture element sizes down to 100 μm [23]. Reproducing the sequence of forces felt by the skin has been proven by prior studies to be sufficient to facilitate the perceptual identification of surface texture [9,14,22,24].

Little is known about how vibrations propagate through the hand. Furthermore, the interpretation mechanism of patterns of vibrations reflecting the nature of the objects touched remains unknown. A recent study analyzed the spatial distribution of vibrations propagating through the skin on the hand during active touch, grasping, and manipulation tasks [25]. It revealed that the vibration patterns evolved rapidly in time and varied systematically with touch interactions, which made it possible to decode the modes of interaction using the touched objects. Hence, vibrotactile information distributed throughout the hand can transmit information regarding objects that are explored and manipulated.

2.2. Types of haptic texture

There is a great variety of textures in the real world. These can be generated haptically according to the types of vibrations produced. The following are the most common types.

2.2.1. Small random textures

Small random textures are the most common form of textures. Most surfaces are not completely smooth but have minute irregularities. Touching even a smooth tabletop with the tip of a pen produces small, almost imperceptible vibrations. A plastic keyboard produces a bit more pronounced vibrations, whereas sandpaper produces vibrations of higher frequency. This type of vibration is easiest to reproduce by adding only a tiny noise value to the computed response force. This can be added following collision detection, and is virtually independent of probe position and direction of motion.

2.2.2. Large random textures

Large random textures, such as gravel or rough stone, are random, but their texture force is dependent on the direction of motion and the position of the probe. These textures can be generated by height maps that add a texture force value depending on the probe position. Solid noise can be used to generate a semi-random texture dependent on the position. If irregularities are large, they must be incorporated into the virtual object mesh producing the actual displacement of the surface. Otherwise, the collision effect is unrealistic [6]. Ideally, graphic and haptic rendering must be synchronized. If they are not aligned, the effect may be lost on the user.

2.2.3. Regular textures

Regular textures are like grids and gratings with a small pattern repeated on the surface. These textures require the calculation of a function or a heightmap. For some textures, the vibration produced is highly dependent on the direction of the probe. However, for very small grid-like regular textures, a mix of sinusoidal functions can also convey the basic characteristics of a texture, as any periodic signals can be represented by a sum of sinusoidal functions [6].

2.2.4. *Flow textures*

Flow textures like fire, water, and wind are not confined to a shape, but can be simulated in an environment or a volume [6]. A noise function can be used to represent such textures. Like large textures, these textures must not be completely random; their randomness needs to evince a pattern.

3. Past work

Efforts to quantify haptic textures began in the early 90s, when haptics was in its infancy. The sandpaper system by Minsky et al. [4] was the first attempt to render algorithmic haptic texturing, and is very similar to bump mapping in graphics. Surface normals are used to compute the haptic forces to be rendered. In their experiments, grids and gratings with varying degrees of roughness were simulated using gradients of pre-existing texture height maps. These techniques required explicit data storage for rendering, and were likely to exhibit noticeable faults at the boundaries of the textured regions [9]. Stochastic models were developed by Siira and Dinesh [5], and Fritz and Barner [6] for haptic textures, since most textures found in the world have probabilistic components. Both these models were able to create mutually differentiable textures using Gaussian noise. They produced textures with varying degrees of roughness by altering the mean, the variance, and the sampling period, although the textures generated were not realistic. Texture vibrations were independent of tool velocities, and the texture force was reduced to zero below a velocity threshold.

Realistic haptic textures were recorded and reproduced by Vasudevan and Manivannan [26]. They recorded force vibrations as the tool moved across a surface with a certain velocity, and used the resulting frequency spectrum to haptically recreate the surface. Their approach can help create realistic textures, but requires sensitive instruments for force measurement. The approach is more suited to small random textures, where the direction and position of the tool are irrelevant. Kuchenbecker et al. worked on modeling and rendering realistic textures from high-frequency tool acceleration signals [7,8,27]. A specific hardware was designed to record data in unconstrained tool-surface interaction in this vein. The study indicated that roughness can be accurately recreated, but no significant success was achieved with regard to the hardness and slipperiness of virtual textures. Moreover, the platform used for this work was incapable of modulating stiffness or friction, and a different rendering platform was needed to fully render and test the textures. Moreover, the algorithm proposed in this study works in 2D space, but has not been tested in 3D environments. Meyer et al. 2016 [9] proposed a procedural texturing method to create haptic textures by recording a series of fingertip swipes across 11 textures and storing the data as spatial friction maps. They analyzed them using a space-frequency transform and proposed a Weibull distribution model as the best-fit representation of the measured data. The method requires instruments and involves several precomputing steps. Moreover, no experiments and subjective evaluations were presented for the haptic textures generated using this method.

Shopf and Olano [28] proposed a framework for procedurally defining haptic texture where textures were created using haptic texture shaders. They claimed to have been able to create textures that can be related to visual textures, though are not necessarily recognizable without the presence of visual cues. Their shaders provide the illusion of surface characteristics by altering previously calculated forces due to object collision in the haptic pipeline. Although the method can provide correspondence between visual and haptic shaders that can allow dynamic modification of the haptic texture according to user interaction, it is still necessary to maintain a one-to-one relationship between the visual and the haptic shaders, which is very difficult to achieve in most cases. Furthermore, because haptic shaders execute following haptic collision, they can only be used to model small-scale surface displacements, whereas large displacements cause inconsistent collisions, as the authors acknowledged. In our work, there is no need for one-to-one correspondence; it can also be applied to large-scale displacements.

This study can be regarded as an extension of our previous work that explored the potential for solid noise-based texture. A range of realistic textures were generated that can be related to graphical texture using a single equation. Furthermore, we found that a diverse range of users could recognize the textures to some extent without any visual cues, and could relate the haptic textures to graphical textures.

4. Haptic texture generation

The first step in texture generation involves considering the mathematical representation of the texture of a real object as the base for the definition of virtual texture. Since the object is touched through a probe, texture information is conveyed through vibrations that have the following characteristics: 1) Texture is perceived only when the probe *moves* across a surface. There is no vibration felt by a stationary probe. 2) The *same* texture is felt at the *same* position. 3) The vibration felt depends on the *velocity*.

Therefore, the virtual texture should also simulate these properties. Texture can be created by making texture variations a part of the shape of the object and displacing the actual surface. The goal of this work is to create a haptic texture that is independent of the object mesh and superimpose a texture force on the response force output. In other words, we used a force perturbation approach instead of actual displacement. Although displacement can negate the effect of force perturbation, as has been shown [4], the perturbation can still provide a satisfactory approximation. Thus, texture can be incorporated without complicating the basic geometry. Texture force can be added to other response forces once collision is detected. The total response can be calculated as:

$$\overrightarrow{F}_{result} = \overrightarrow{F}_{contact} + \overrightarrow{F}_{friction} + \overrightarrow{F}_{texture}$$

Where $\overrightarrow{F}_{texture}$ is the force determined by the haptic texture and $\overrightarrow{F}_{friction}$ is the force due the surface friction and $\overrightarrow{F}_{contact}$ is the contact force based on the collision and penetration distance. The equation was described in many references [6,29–31].

4.1. *Solid noise approach*

In order to design an algorithm for texture force perturbation, a survey of the nature of haptic texture through a tool was conducted. It was observed that most textures are random in nature. Therefore, a procedural approach involving a noise function was considered the best to implement a fast and diverse texture algorithm capable of representing a wide variety of textures.

The coherent noise function [32] was found the most suitable for implementing our approach because it is a controllable and semi-random noise function that can be exploited to create diverse textures. The same inputs result in the same outputs, and a change in the output corresponds to that in the input. Therefore, a 3D coherent noise function in place of the probe as input has all the required characteristics. As solid noise, the resulting texture is independent of the object's geometry, and does not need to be mapped to a surface. Coherent noise is a continuous function providing an appropriate value for each haptic cycle. If the position of an object is the input for the function, for no motion will there then will be any vibration. In other words, texture force is constant for a stationary tool. A vibration is only produced when there is a change in position, and automatically depends upon velocity. Each bump and pit remain fixed in position. The resulting texture also exhibits neighborhood dependencies. Thus, it can be concluded that texture based on a

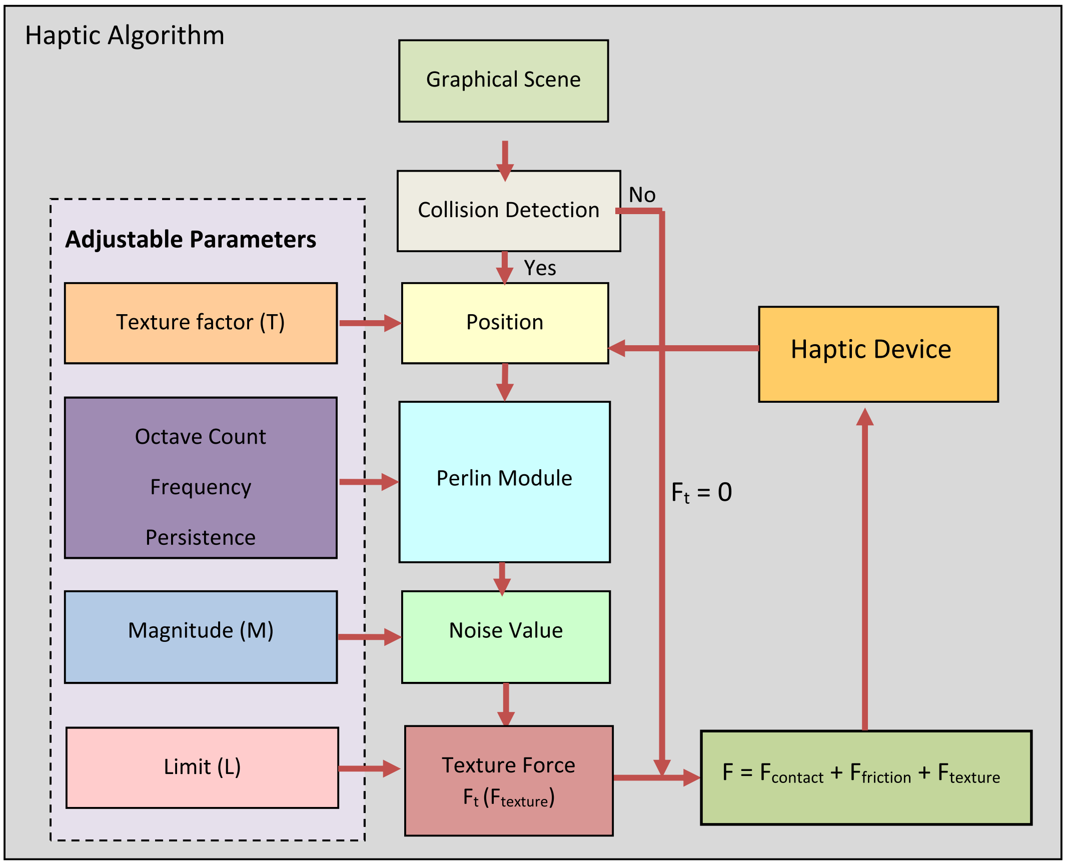


Fig. 1. The algorithm for generating texture force using Perlin noise and the adjustable parameters used to tune the resultant texture force.

solid coherent noise function can meet all requirements without requiring additional computation.

Perlin noise is a special type of solid coherent noise that combines a set of random coherent noise functions of varying frequencies and magnitudes [32]. Perlin noise can provide a set of parameters to produce mutually differentiable textures that can be aligned to a graphical texture. Perlin noise has been used extensively for graphical textures but its use for haptic texture has been limited. Although this use has been hinted at before in [5,6], it had only been used then for haptic texture shaders [28], and the use was limited. They implemented five shaders for five textures.

Perlin noise is generated with the fractal summation of basic noise functions. Perlin in [33] introduced a method where each noise is multiplied by a weight controlling its contribution to the final result. The equations for generating the final noise patterns are as follows:

$$n_p(x,y,z) = \sum_{i=0}^{k} \frac{n_i(x,y,z,p.(1/a)^i)}{b^i} \quad (1)$$

$$n_p(x,y,z) = \sum_{i=0}^{k} n_i(x,y,z,p.H^i)\cdot H^i \quad (2)$$

$$n_p(x,y,z) = \sum_{i=0}^{k} n_i(x,y,z,p.(1/lac)^i)\cdot per^i \quad (3)$$

Where $n_p(x,y,z)$ is Perlin noise and $n_i(x,y,z)$ is basic noise i. The parameter a defines the irregularity of noise. The parameters a and b usually are set to be equal. In this case, they can be replaced by $1/H$. Each noise $n_i(x,y,z)$ is called an octave and usually formed by a twofold increase in frequency and twofold decrease in amplitude in Perlin noise. Frequency is an input parameter to control the number of cycles per unity length and is calculated as $1/(p.H^i)$. This means that for each octave the frequency will increase by $1/H$, which is controlled by a multiplier to determine how quickly the frequency increases for each successive octave and is called lacunarity. Finally, we have the persistence parameter which determines how quickly the amplitude for each successive octave will diminish. Usually the value H is used to decrease the amplitude so we can get the equation (3). The constant k determines how many octaves you want to generate and combine for final Perlin noise. More details of how each parameter affects the final signal and samples of signals can be found in the libnoise library site that was used in this project [32].

In the algorithm proposed here, the Perlin noise equation (equation (3)) with additional parameters (equation (4)) is used for many differentiable textures. This fact not only makes it easy to incorporate our algorithm in any existing code but also renders it easy to make changes at runtime. Altering only the values of the parameters while using the same function can result in differentiable textures. Thus, the proposed algorithm is simple yet powerful.

4.2. *Haptic texture implementation*

The haptic device we used for implementation was PHANToM Omni from SensAble [34]. A 2.5 GHz dual-core Athlon system was used with C++ for programming and Chai3d libraries [35] for haptics and libnoise to generate Perlin noise [32].

The equation (3) for generating Perlin noise is used with additional parameters to gain more control on the final generated texture as shown in Fig. 1. The final equation for generating the texture force is as follows:

$$F_{texture}(x,y,z) = L\cdot\sum_{i=0}^{k} M\cdot[n_i(T.Pos_{x,y,x},p.(1/lac)^i)\cdot per^i] \quad (4)$$

Where L is the limit factor to control the maximum texture force, T is the texture factor to adjust the force based on the position, M is the magnitude to control the final strength of the force after generated by the Perlin module. Other parameters in equation (4) (frequency,

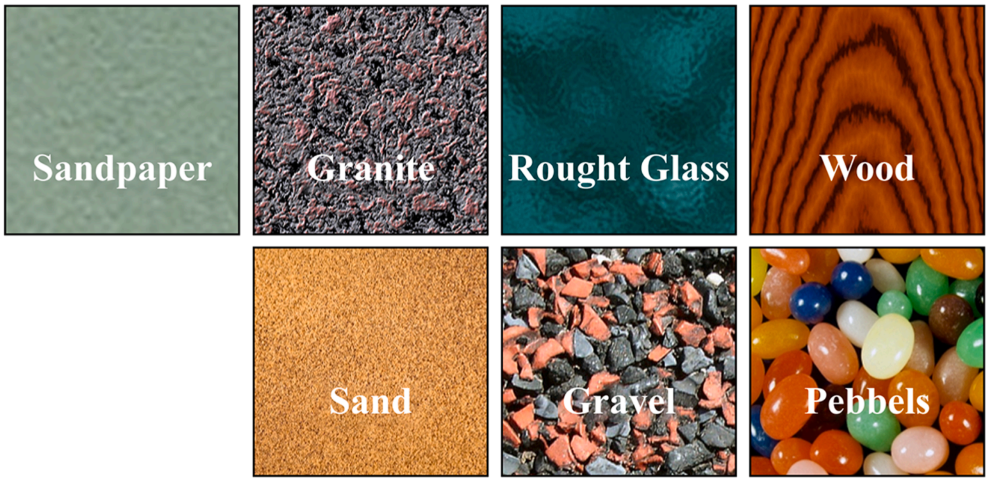


Fig. 2. The seven textures that were haptically modeled using Perlin noise. Each surface was tuned to induce unique vibrations that represent the microscopic details.

lacunarity, persistence) the same in equation (3) which are already explained.

The input to the noise function was the tool position multiplied by a texture factor that could be controlled by a user, thus mapping tool position to texture space. The noise value returned was multiplied by a magnitude factor that could be changed to produce textures with different maximum feature sizes. The number of octaves could make the texture smooth, or harsh and rough. A greater magnitude of frequency meant closer spacing of irregularities, this complies with the duplex theory about the need to consider vibrational and spatial stimulation as mentioned in [23]. Finally, the persistence factor determined whether to suppress or emphasize higher-frequency octaves. A small persistence value means that higher octaves will be suppressed in the final texture. While a high persistence value makes the higher octaves more pronounced. Texture factor and limit factor were also used for additional control on texture shape. The algorithm for generating the texture force with the parameters used is illustrated in (Fig. 1).

A texture force was returned only when a tool penetrated an object. It was then added to the contact and the friction forces that had been previously calculated. The texture force was produced only parallel to the x-axis. It has been mentioned in past work that the texture force should be proportional to the contact force [5,6] and tangential to the surface. For this work, the approach involved performing as small a number of computations as possible and not complicate the equation unless needed. These additions can be easily implemented, but the performance was found to be acceptable even without these additions to the texture equation. There was no difference perceived between texture forces being tangent or normal to the surface. As solid noise is dependent on position, the resulting textures were automatically proportional to velocity.

Making the noise function dependent on object position made the haptic texture more realistic as the vibrations produced were fixed in space. The vibrations successfully mimicked an actual tool moving along a surface with a certain velocity. Having tailored the Perlin noise function somewhat similarly to a texture, it was as if every irregularity of the graphical texture was felt even though there was no true relation to it.

Several textural effects could be added without affecting the final rendering performance. This proved that calculation time was sufficiently small not to affect the haptic update rate. The time taken by a haptic cycle using the Perlin texture was between 20 μs and 40 μs. Higher-frequency textures were observed to take longer, with a comparatively longer cycle once in a while. However, on the whole, the update rate seemed to be satisfactory and the response stiff.

Table 1
Different textures.

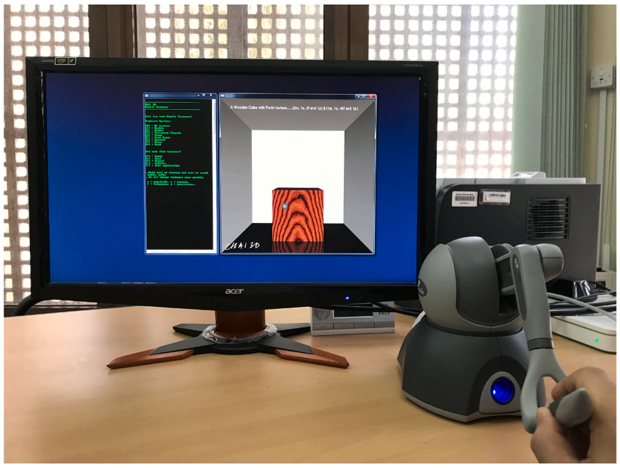
Texture	Mag.	Octave	Frequency	Persistence
Sand Paper	3	1	50	1
Granite	6	2	1	0.5
Rough Glass	3	1	4	1
Wood	1	1	40	1
Sand	2	2	10	0.5
Gravel	6	2	5	0.5
Pebbles	8	2	1	0.5

4.2.1. *Texture generation*

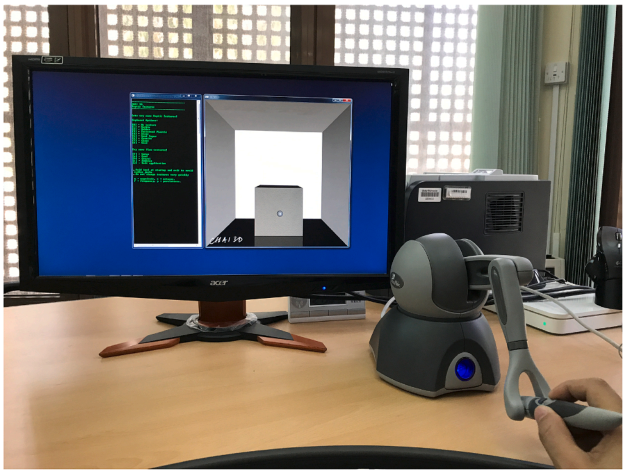
The Perlin noise libraries selected for noise generation provided many interesting features that could be used to create a great number of diverse textures. A basic noise function with few octaves was used in the project as the time taken for calculation was also crucial. Even for a basic Perlin module, several features could be manipulated to provide mutually differentiable texture. These are the number of *octaves*, *frequency*, and *persistence* in addition to introduced *texture* and *limit* factor to gain more control on the texture shape as explained in (Fig. 1). Seven common and diverse materials were selected with different hardness, roughness, slipperiness, and fineness for our set of textures to be implemented and tested: sandpaper, granite, textured glass, wood, sand, gravel, and pebbles (Fig. 2). The parameter values were tuned and carefully adjusted to match the real feeling of the real textures during the development phase. Ten subjects including the authors were involved in tuning the generated textures with the real texture. The following are some examples of how parameter values were chosen for the textures generated:

- The *sandpaper* texture was generated simply by using an octave of coherent noise with high frequency and a moderate magnitude resembling random noise.
- The *granite* texture featured high magnitudes and more octaves as the texture pattern became larger.
- *Wood* had one octave of a low-frequency corresponding to rings, and another at high frequency to represent grain. Two modules were used to achieve the effect.
- The *rough glass* texture had very small friction and a single octave of noise.
- Higher-frequency textures resulted in the *sand and gravel* textures.
- A high-magnitude noise function in a viscous cube resulted in a *pebble*-like effect.

Table 1 shows the values selected for the experiment.



(a)



(b)

Fig. 3. Experimental setup for haptic texture evaluation. The subject sat in front of a table equipped with a PHANToM haptic device and a monitor to see different cubes with different textures. The cubes were presented in two modes, with visual graphical textures that representing the real texture (a) and without visual graphical textures of the haptic texture (b).

5. Evaluation

It was evident at once that the textures were generated successfully and were mutually differentiable. After some exposure, the textures were immediately recognizable. Even though there was no true relation between the graphical and haptic textures, humans were able to relate them, and both textures were felt to be mutually correspondent. A series of experiments were then conducted to verify and evaluate how well the textures could be perceived by a variety of users. The aim of the experiment was not to accurately detect texture properties, but a subjective evaluation to determine how well ordinary users could identify and evaluate texture, and whether they perceived the given textures as realistic, differentiable, and correspondent to features of a graphical texture. Haptic simulation should provide a realistic experience that can be felt by any person without the need for previous exposure to the technology. Thus, it can enhance the haptic experience.

5.1. Experiment setup

The experiment was conducted with the help of 20 volunteers from different backgrounds and academic levels. They included males, females, left-handed people, and right-handed people. All volunteers had had either no or negligible past experience working with a haptic device.

All subjects sat at a desk with a PC display in front of them and used a PHANToM haptic device to interact with a virtual world that consists of a virtual cube placed at the center of a virtual room. The subjects could interact in this 3D environment and touch the cube from any direction. The experimental setup is shown in Fig. 3 (a). Different textures were generated and rendered, as shown in Table 1.

The experiment was conducted in two modes; the haptic-visual mode and the haptic-only mode. In the haptic-visual mode, the setup is shown in Fig. 3(a) was used. Here, the users could experience the full range of textures and evaluate them with the help of a questionnaire. For the haptic-only mode, the setup is shown in Fig. 3(b) was used. The user could not see the graphical textures but tried to recognize a texture using only haptic cues. The haptic textures were presented one by one in random order for the user to recognize.

5.2. Experimental procedure

The subjects provided their informed consent to the experiment and were briefed about its objective. To familiarize them with the haptic device and the environment, we first allowed them to touch a smooth texture, and observe the response as well as the basic device operation. The experiments were conducted in two phases: the haptic-visual phase, and the haptic phase. The phases were presented randomly to subjects to avoid bias to the haptic visual phase as the subjects may tend to create a visual-haptic association that may produce considerable bias in the haptic response.

5.2.1. Realism and graphical correspondence ratings

In this phase of the study, the environment represented the virtual cubes textured with the graphical image representing the texture of each material (see Fig. 3 (a)). The textures were presented in randomized order, and the subject was asked to freely explore the virtual surface while judging how well it matched the sensation of the corresponding real material. The subject then rated the "realism" of each texture based on the statement: "The virtual and real texture are completely indistinguishable." The subjects were then asked to judge the relationship between the feeling of the texture and the graphical texture representing it. The subject rated the correspondence between the graphics and the feeling based on the statement: "The graphical texture completely corresponds to the feeling, and accurately represents the virtual texture."

5.2.2. Recognition accuracy

In this experiment, the subjects were instructed that their main task was to determine the type of virtual texture from the set of seven generated textures shown in Fig. 2. The cubes were presented to the subjects without accompanying graphical texture to represent the texture of the real materials; instead, a solid color was used to represent all virtual textures, as shown in Fig. 3(b). Therefore, the subjects had to rely solely on their haptic sensations to determine the type of virtual texture.

6. Results

In this section, we analyze the recognition accuracy of each simulated texture, the realism ratings, and the correspondence between the graphics and the sensation provided by subjects during the experiment.

6.1. Recognition accuracy

For the calculation of recognition accuracy, the input from each subject was "100" for correct recognition, and "0" for incorrect recognition. The mean value of all subjects' inputs for each texture was calculated to represent the recognition accuracy for each texture, as

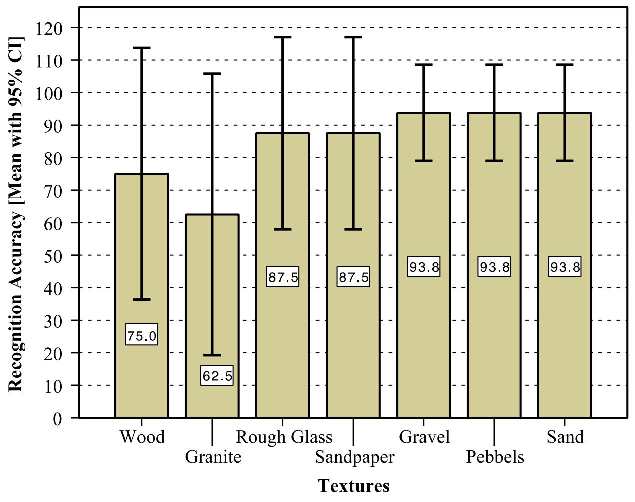


Fig. 4. Recognition accuracy for each virtual texture.

shown in Fig. 4. The average recognition accuracy for all subjects for all textures was 84.8%.

6.2. Realism and graphical correspondence ratings

Fig. 5 and Fig. 6 show the mean and standard deviation, respectively, for the realism and correspondence ratings of each virtual texture. Figs. 5 and 6 were organized in the same order as Fig. 4 for the sake of comparison and a boxplot is selected for containing several measures. The bottom of the box indicates the 25th percentile and the top of the box represents the 75th percentile which means Twenty-fine percent of cases have values above the 75th percentile which means that 50% of the cases lie with the box. 95% of the data approximately are expected to lie between the whiskers of the data are distributed normally. The circle points are outliers and asterisks are extreme outliers. Fig. 5 shows that all textures were more than 85% realistic. The recognition rate for gravel and pebbles was among the highest at 93.8%, and their realism was above 90%. The correspondence in Fig. 6 shows the same tendency as the realism in the sense that all textures scored more than 85%.

To study the impact of realism and correspondence on recognition rate, a two-way analysis of variance (ANOVA) at 0.05 significance was performed. Realism was found to have no statistically significant effect on recognition at (F = 0.229, $p = 0.948$). Similarly, the correspondence factor was found to have no significant effect on recognition as well at (F = 0.102, $p = 0.991$). Moreover, there was no interaction between the two factors of realism and correspondence (F = 0.199, $p = 0.976$). This showed that neither realism nor correspondence has any impact on recognition accuracy. This proves that the identification of each texture in the recognition experiment was based solely on the haptic sensation, and exposure to virtual texture with the visual aid did not influence subjects' decisions.

7. Discussion

The method proposed in this work achieved high average recognition accuracy of 84.8% for all seven textures tested and when subjects were presented with them one by one in a random sequence.

The results were considered satisfactory due to the following observations: The textures could be felt and identified by users without previous exposure to haptic devices. All participants reported immediately being able to feel the texture, and the textures were differentiable.

Some textures were more difficult to tell apart, but this was still possible as more than 70% guessed correctly, except for wood and granite, which had a lower percentage of recognition. This could have obtained because the textures themselves may be hard to tell apart using a tool, without feeling the surfaces or being able to see them.

Another reason for this could have been that the parameter values selected may need further improvement to correspond more closely to the texture. Another point to note from Table 1 is that textures such as wood, textured glass, and sandpaper, all had the same magnitude, which might also have contributed to mistaken identification.

It was also interesting to note that some textures received a high rating and were described to be similar to the respective graphical texture, while their percentage of correct identification was low. This could have been because these textures were simulated realistically, but were not distinguishable when presented in random sequence.

The participants also responded to some questions regarding the ease or difficulty of guessing the textures following the experiment. They ascribed errors to three causes: 1) the textures selected were difficult to distinguish in real life as well as in experiments; 2) the textures were not simulated realistically; 3) they were unfamiliar with the device or the textures. Fig. 7 shows the results of the reasons for subjective error.

It is worth mentioning that we experimented with different haptic devices to confirm that the proposed algorithm is general and works on any haptic device. This was achieved easily because the Chai 3D library supports many haptic devices from different companies like Force

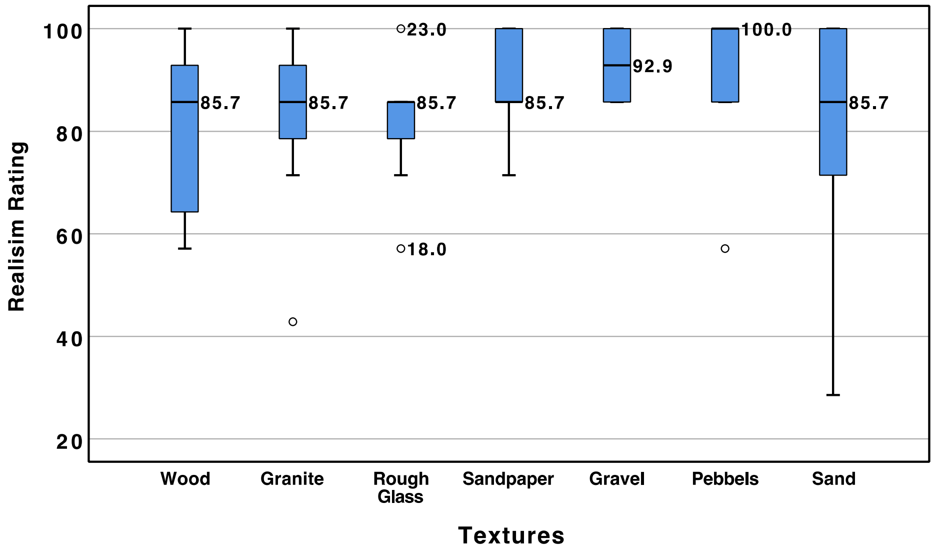


Fig. 5. Realism rating for each texture assigned by subjects. The values represens the median.

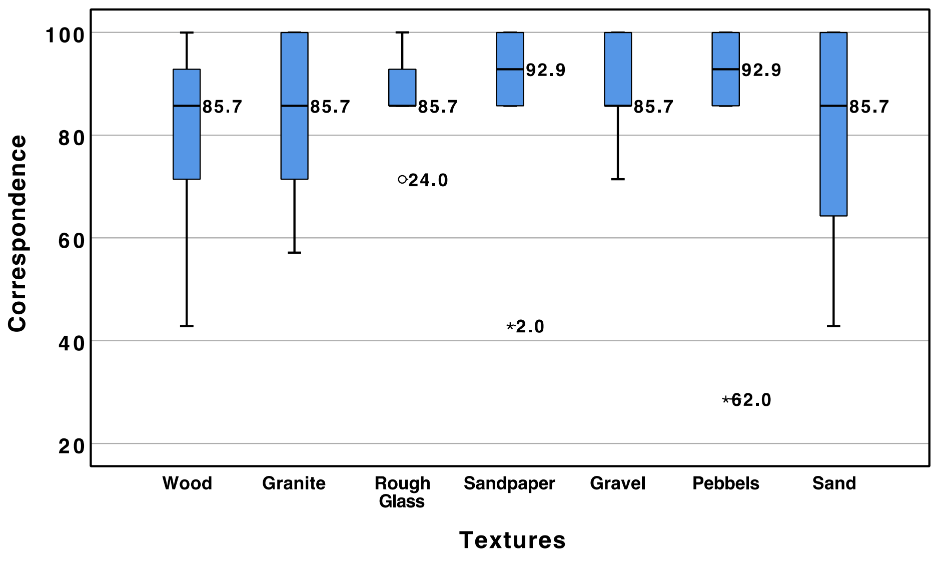


Fig. 6. Correspondence ratings between the haptic sensation for each texture and its graphical texture to show how well the graphics represented the real texture, and how this affected recognition accuracy.

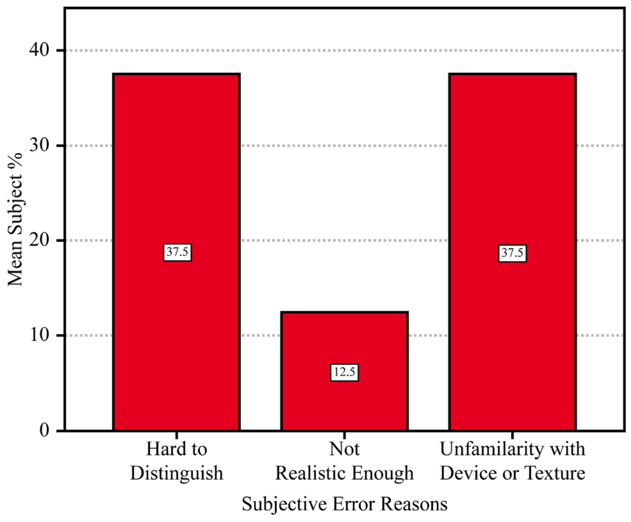


Fig. 7. The subjects ascribed the error in guessing the textures correctly to the three reasons illustrated here.

Dimensions, Novint Technologies, and 3DSystems without the need for any modification in the code. We have tried with Falcon haptic device from Novint Technologies which is considered a consumer-level alternative and lower grade than the Phantom haptic device, see Fig. 8. The experiment went very well without any issues related to vibration since the haptic rendering update was >1 kHz so smooth haptic rending was achieved without any vibration or buzzing. The only difference is the users felt that the quality of force feedback is better in Phantom which gives a better sensation of the surfaces, however, no issues related to vibrations. This result confirms the study presented in [36] where it was found that Falcon has higher damping than the Phantom. As a result, this allows higher force amplitudes in Falcon and decreases the correct discrimination of the applied forces. However, the objective and subjective evaluation found no significant difference between the devices in terms of task completion time. This also demonstrates that the presented result in this study applied for different haptic devices. Nevertheless, the quality of the haptic feeling may slightly affect the recognition accuracy.

8. Conclusions

In this paper, we proposed a simple, general, and customizable method for creating haptic virtual textures using simple Perlin solid noise that can be easily integrated into any code and used by any haptic device.

The Haptic texture is customizable through magnitude, frequency, and persistence parameters. The position of the haptic tool can be used as an input to the noise function multiplied by a factor to achieve a mapping between the tool and the texture space. Then, the noise value is multiplied by a magnitude factor that can be changed to produce textures with different maximum feature sizes. We can control also the number of octaves to make a texture smooth, or harsh and rough. A greater magnitude of frequency would mean closer spacing between irregularities, and persistence would determine whether to suppress or emphasize higher-frequency octaves. The method also allows a user to change the texture at runtime and can be incorporated into any existing code, thus making it general in terms of its applicability. Our code can be used with other haptic devices without the need for any changes, due to the open-source Chai 3D library. Moreover, the solid noise texture is independent of object geometry, and can be applied to any shape without additional computation.

We conducted a study to evaluate the effectiveness of the proposed approach based on subjects' ability to guess virtual textures without any visual or audio aid. We also evaluated the realism of our virtual textures, and the correspondence between haptic feeling and graphical texture, and showed that the recognition accuracy of the proposed method was sufficient, since realism and correspondence had no influence on recognition rate, and the subjects' determinations depended solely on their haptic sensations in guessing virtual texture.

The proposed algorithm is like a tool that can be used to simulate any texture. The power of the algorithm also depends on how it is used. Correct values of parameters can convey a texture more accurately. To the best of our knowledge, haptic texture has not been simulated by a similar algorithm. The textures simulated could not only be made rough or smooth, but could also be made to represent real textures.

Texture implementation was sufficiently fast to be added to any existing program without any appreciable slowing of the update process. No delay or vibration was noticed in the haptic cycle, as texture rendering and stiffness response did not witness any noticeable degradation.

Different textures can be easily chosen, reproduced, and even created

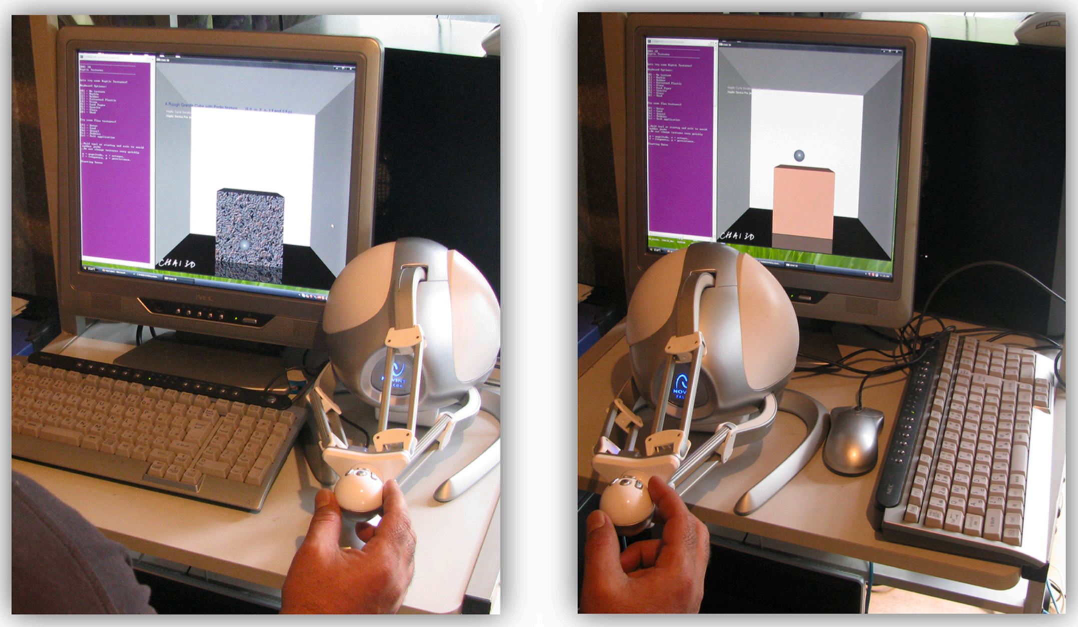


Fig. 8. Experimental setup for haptic texture with Falcon haptic device for different textures.

at runtime by a user in our method. Every object in a haptic scene can have a unique haptic texture without complicating the overall geometry, as the texture algorithm is applied following collision detection such that only the texture of the object touched is returned at a time. Unlike graphical textures, haptic textures do not affect the update rate, since only one point is returned at a time. The texture of only the object or the surface touched needs to be computed and returned. A number of parameters—magnitude, persistence, frequency, number of octaves, and maximum limit—can be used to generate different textures. The proposed method is a simple, compact, and versatile approach to haptic texture generation that can be implemented on common desktops and similar devices.

9. Future work

A simple Perlin texture effect was created, but it could be a gateway to unlimited possibilities. It may be possible to use the same algorithm to design such environments where a user can change texture parameters directly or alternately the parameters can be made to depend on some other event or data. Similar effect classes can be created using more advanced libnoise modules. As graphical textures can also be created using the same functions so both graphic and haptic textures can be made to correspond exactly. Haptic texture uses the coordinates of an object in a workplace to compute the solid noise value. So if a graphical texture is also generated in such a way as to align the graphical texture space with the haptic texture space then both textures can be perfectly aligned. Therefore, every irregularity seen in graphical texture can also be made haptically palpable although haptic texture should be a simpler version of its graphical counterpart. Motion can be incorporated by making the haptic texture dependent upon object local coordinates then it will remain the same even if the object is moving around and changing position in global coordinates. Nevertheless, the values used in generating the virtual textures were not derived from statistical data of actual textures but were rather experimentally derived from the visual textures available. Therefore, better approximations can be achieved from actual data that can yield much better performance.

CRediT authorship contribution statement

Osama Halabi: Supervision, Conceptualization, Methodology, Formal analysis, Visualization, Writing - review & editing. **Gulrukh Khattak:** Software, Investigation, Methodology, Data curation, Validation, Writing - original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper was jointly supported by Qatar University M-QJRC-2020-7. The findings achieved herein are solely the responsibility of the authors. The Open Access funding is provided by the Qatar National Library.

We would like also to thank Francois Conti from Stanford University on advice on how to use PHANToM with CHAI 3D.

References

[1] M.M. Taylor, S.J. Lederman, Tactile roughness of grooved surfaces: A model and the effect of friction, Percept. Psychophys. (1975), https://doi.org/10.3758/BF03203993.
[2] R.L. Klatzky, S.J. Lederman, C. Hamilton, M. Grindley, R.H. Swendsen, Feeling textures through a probe: Effects of probe and surface geometry and exploratory factors, Percept. Psychophys. (2003), https://doi.org/10.3758/BF03194587.
[3] R.L. Klatzky, S.J. Lederman, Tactile roughness perception with a rigid link interposed between skin and surface, Percept. Psychophys. 61 (1999) 591–607, https://doi.org/10.3758/BF03205532.
[4] M. Minsky, O. Ming, O. Steele, F.P. Brooks, M. Behensky, Feeling and seeing: issues in force display, ACM SIGGRAPH Comput. Graph. 24 (1990) 235–241, https://doi.org/10.1145/91394.91451.
[5] J. Siira, D.K. Pai, Haptic Texturing-A Stochastic Approach, n.d.
[6] J.P. Fritz, K.E. Barner, Stochastic models for haptic texture, 1996. https://doi.org/10.1117/12.263011.
[7] J.M. Romano, K.J. Kuchenbecker, Creating realistic virtual textures from contact acceleration data, IEEE Trans. Haptics. (2012), https://doi.org/10.1109/TOH.2011.38.

[8] H. Culbertson, J. Unwin, K.J. Kuchenbecker, Modeling and rendering realistic textures from unconstrained tool-surface interactions, IEEE Trans. Haptics. (2014), https://doi.org/10.1109/TOH.2014.2316797.
[9] D.J. Meyer, M.A. Peshkin, J.E. Colgate, Tactile Paintbrush: A procedural method for generating spatial haptic texture, IEEE Haptics Symp. HAPTICS (2016), https://doi.org/10.1109/HAPTICS.2016.7463187.
[10] M. Wiertlewski, C. Hudin, V. Hayward, On the 1/f noise and non-integer harmonic decay of the interaction of a finger sliding on flat and sinusoidal surfaces, in: 2011 IEEE World Haptics Conf. WHC 2011, 2011,: pp. 25–30. https://doi.org/10.1109/WHC.2011.5945456.
[11] W. Hassan, A. Abdulali, S. Jeon, Authoring New Haptic Textures Based on Interpolation of Real Textures in Affective Space, IEEE Trans. Ind. Electron. 67 (2020) 667–676, https://doi.org/10.1109/TIE.2019.2914572.
[12] R.H. Osgouei, J.R. Kim, S. Choi, Data-Driven Texture Modeling and Rendering on Electrovibration Display, IEEE Trans. Haptics. 13 (2020) 298–311, https://doi.org/10.1109/TOH.2019.2932990.
[13] A. Abdulali, I.R. Atadjanov, S. Jeon, Visually Guided Acquisition of Contact Dynamics and Case Study in Data-Driven Haptic Texture Modeling, IEEE Trans. Haptics. 13 (2020) 611–627, https://doi.org/10.1109/TOH.2020.2965449.
[14] M. Janko, R. Primerano, Y. Visell, On Frictional Forces between the Finger and a Textured Surface during Active Touch, IEEE Trans. Haptics. (2016), https://doi.org/10.1109/TOH.2015.2507583.
[15] M.D.R.R. Minsky, Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-feedback Display, Massachusetts Institute Technology (1995).
[16] D.C. Ruspini, K. Kolarov, O. Khatib, The haptic display of complex graphical environments, in: Proc. 24th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '97, 1997. https://doi.org/10.1145/258734.258878.
[17] C. Ho, C. Basdogan, M. Srinivasan, A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments, ASME Winter Annu. Meet. 61 (1997) 77–84. http://www.rle.mit.edu/touchlab/publications/1997_001.pdf.
[18] M.A.M. Costa, M.M.R. Cutkosky, S. Lau, Roughness perception of haptically displayed fractal surfaces, Proc. ASME IMECE DSC Haptics Symp. (2000) 1–7 (Costa).pdf, http://ti.arc.nasa.gov/m/pub-archive/178h/0178.
[19] M.A. Lawrence, R. Kitada, R.L. Klatzky, S.J. Lederman, Haptic roughness perception of linear gratings via bare finger or rigid probe, Perception. 36 (2007) 547–557, https://doi.org/10.1068/p5746.
[20] L. Skedung, K. Danerlv, U. Olofsson, C. Michael Johannesson, M. Aikala, J. Kettle, M. Arvidsson, B. Berglund, M.W. Rutland, Tactile perception: Finger friction, surface roughness and perceived coarseness, Tribol. Int. 44 (2011) 505–512. https://doi.org/10.1016/j.triboint.2010.04.010.
[21] M. Wiertlewski, J. Lozada, E. Pissaloux, V. Hayward, Causality inversion in the reproduction of roughness, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2010: pp. 17–24. https://doi.org/10.1007/978-3-642-14075-4_3.
[22] M. Wiertlewski, J. Lozada, V. Hayward, The spatial spectrum of tangential skin displacement can encode tactual texture, IEEE Trans. Robot. 27 (2011) 461–472, https://doi.org/10.1109/TRO.2011.2132830.
[23] M. Hollins, S.R. Risner, Evidence for the duplex theory of tactile texture perception, Percept. Psychophys. (2000), https://doi.org/10.3758/BF03206916.
[24] D.J. Meyer, M.A. Peshkin, J.E. Colgate, Modeling and synthesis of tactile texture with spatial spectrograms for display on variable friction surfaces, IEEE World Haptics Conf. WHC 2015 (2015) 125–130, https://doi.org/10.1109/WHC.2015.7177702.
[25] Y. Shao, V. Hayward, Y. Visell, Spatial patterns of cutaneous vibration during whole-hand haptic interactions, Proc. Natl. Acad. Sci. 113 (2016) 4188–4193, https://doi.org/10.1073/pnas.1520866113.
[26] H. Vasudevan, M. Manivannan, Recordable haptic textures, in: Proc. 2006 IEEE Int. Work. Haptic Audio Vis. Environ. Their Appl. HAVE 2006, 2007. https://doi.org/10.1109/HAVE.2006.283779.
[27] H. Culbertson, J.M. Romano, P. Castillo, M. Mintz, Refined Methods for Creating Realistic Haptic Virtual Textures from Tool-Mediated Contact, Proc. IEEE Haptics Symp. (2012) 385–391.
[28] J. Shopf, M. Olano, Procedural Haptic Texture (2006).
[29] O. Halabi, Y. Halwani, Design and Implementation of Haptic Virtual Fixtures for Preoperative Surgical Planning, Displays (2018), https://doi.org/10.1016/j.displa.2018.07.004.
[30] O. Halabi, H. Kawasaki, Five Fingers Haptic Interface Robot HIRO: Design, Rendering, and Applications, Adv. Haptics, InTech (2010) 221–240, https://doi.org/10.5772/8691.
[31] H. Kawasaki, Y. Ohtuka, M.O. Alhalabi, T. Mouri, Haptic rendering and perception of frictional moment, Proc. EuroHaptics. (2006) 201–206.
[32] J. Bevins, libnoise, 2016. http://libnoise.sourceforge.net.
[33] K. Perlin, Improving noise, in: Proc. 29th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '02, ACM Press, New York, New York, USA, 2002: p. 681. https://doi.org/10.1145/566570.566636.
[34] T.H. Massie, J.K. Salisbury, The PHANTOM Haptic Interface : A Device for Probing Virtual Objects, ASME Winter Annu. Meet. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. (1994). https://doi.org/http://doi.acm.org/10.1145/1029632.1029682.
[35] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, E. Vileshin, J. Warren, O. Khatib, K. Salisbury, The CHAI libraries, in: EuroHaptics 2003, 2003, pp. 496–500.
[36] L. Vanacken, J. De Boeck, K. Coninx, The Phantom versus the Falcon: Force feedback magnitude effects on user's performance during target acquisition, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6306 LNCS (2010) 179–188. https://doi.org/10.1007/978-3-642-15841-4_19.