An Experimental Study of Perceived Instability During Haptic Texture Rendering: Effects of Collision Detection Algorithm

Seungmoon Choi and Hong Z. Tan
Haptic Interface Research Laboratory
Purdue University
1285 EE Building, West Lafayette, IN 47907-1285
{*chois, hongtan*}*@purdue.edu*

Abstract

This paper presents a quantitative characterization of the instability that a human user often experiences while interacting with a virtual textured surface rendered with a force-reflecting haptic interface. Psychophysical experiments were conducted to measure the maximum stiffness under which virtual textured surfaces were perceived to be stable in a variety of conditions differing in texture model parameters, rendering method, and exploration mode. Unlike our previous study that used a collision detection algorithm with an inherent step change in force magnitude near the textured surface boundary [2], these experiments used an algorithm proposed by Ho et al. [8] that produces continuously changing force magnitudes at the cost of increased computational complexity. We found that the stiffness thresholds resulted from the collision detection algorithm of Ho et al. were not always higher than those obtained with the algorithm used in [2]. The stiffness thresholds depended on the texture rendering method and the exploration mode used in the experiments. We also discuss the types of instability experienced by the subjects and the corresponding characteristics of the proximal stimuli that invoked the perception of each type of instability. With this knowledge, our future work will investigate techniques to mitigate the problem of perceived instability.

1. Introduction

Haptic texture rendering is a growing research field that holds much promise for enriching the sensory attributes of objects in a virtual environment and for allowing precise and systematic control of textured surfaces for psychophysical studies. Our long-term research objectives are to have a better understanding of how to characterize surface textures in physical and perceptual spaces and to develop procedure-based rendering algorithms that can effectively span the human perceptual space for texture. In the past few years, we have been using a force-reflecting haptic interface to study haptic texture perception and rendering (see [2] for a discussion on why force-feedback displays are suitable for rendering surface textures).

The development of computational methods for texture rendering has received much attention from the haptics research community in the past decade (see, for example, [14][12][8][5][16][4][15]; see also [2] for a review). The first problem that we have encountered in implementing the computational methods is that of perceived instability experienced by a human user while interacting with a virtual textured surface. By *perceived instability*, we refer to all unrealistic sensations (such as buzzing or apparent aliveness of a surface) that cannot be attributed to the physical properties of the textured surface. Our work has therefore concentrated on the quantitative characterization of perceived instability (i.e., unrealistic sensations) during haptic texture rendering and the investigation of the sources of such instability, with a goal to explicitly specify and to enlarge the conditions under which haptically rendered textures are guaranteed to be perceptually "clean" (i.e., free of artifacts) and stable.

We conceptualize two major sources of perceived instability during haptic texture rendering: unstable control of the haptic interface and/or improper environment model. In general, the stability of the haptic interface in a control sense is a necessary condition for realistic rendering. Unrealistic sensations such as buzzing and chattering are likely to be due to control instability. Factors such as unbounded behavior of a controller, unmodeled dynamics of a haptic interface, quantization noise of encoders, energy instilling effects of a zero-order-hold, and asynchronous switch time can all lead to control instability [6]. Realistic rendering also requires the environment model to faithfully follow the physical properties of the objects being rendered. In prac-

Proceedings of the11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)
0-7695-1890-7/03 $17.00 © 2003 IEEE

tice, however, the environment model is usually an approximation to the underlying physics with the goal to induce a target percept such as the hardness of a wall or the texture of a surface. To the extent that such simplification introduces artifacts in rendering, the user can perceive a virtual object or surface to be unrealistic or unstable. Therefore, any study of perceived instability has to take into account the effect of both control stability and environment modeling.

Most studies on the stability of haptic interaction based on control theory have focused on the "virtual wall" problem (see, for example, [6][13][1][7]). These studies usually assume that the haptic interaction is of one degree-of-freedom (DoF) and is perpendicular to the wall, the virtual environment dynamics (the wall) is a mechanical second-order linear system (with stiffness and damping parameters), the contact occurs at a point, and the force-feedback device has relatively simple dynamics along the direction of penetration (linear dynamics in most cases). Two main approaches have been taken. The first one explicitly models the dynamics of repetitive human motions that are typically observed in haptic interactions and to use the model as plant dynamics (for example, see [6]). The other approach uses passivity-based control that does not require the explicit models of human user motions [1][13][7]. It assumes that the human arm is passive when coupled to a passive external system [7], an assumption that finds its support in [9]. To the best of our knowledge, no study has extended the aforementioned work to the stability of haptic texture rendering. To do so requires several major extensions to the techniques used for the virtual wall problem. For example, haptic interaction has to be modeled as multi DoFs, environmental dynamics that is much more complex than that of a flat wall has to be considered, contacts along a curve has to be calculated, and force-feedback devices with at least two DoFs will be needed. These extensions will likely lead to a significant increase in the complexity of the theoretical analysis of the stability of a haptic texture rendering system.

In our previous studies [2][3], we took an experimental approach to investigate the issue of perceived instability. We first measured the maximum surface stiffness under which textures rendered with a force-feedback device (PHANToM) were perceived to be stable [2]. Human subjects interacted and judged the perceived stability of virtual textured surfaces rendered under a variety of conditions. The effects of many factors such as parameters of a sinusoidal texture model, texture rendering method, and exploration mode on perceived instability were investigated. We found that the measured parameter space for perceptually stable haptic textured surfaces was quite small. This result significantly restricted the types of surface textures that can be properly rendered in a virtual environment or used in a psychophysical study. We also concluded, through measurements of proximal stimuli (position, force, and acceleration), that perceived instability was mainly due to noise around 192–240 Hz [3]. This frequency region was found to be perceptually distinct from that responsible for encoding texture information (around 26-65 Hz). In addition, we measured the frequency response of the PHANToM, and suggested that the perceived instability was caused by the high-frequency dynamics of the PHANToM (mechanical resonance).

One issue that required further study is the effect of collision detection algorithm on perceived instability of haptic textures. In our previous studies [2][3], collision detection was performed by testing whether the tip of a stylus was inside a flat wall underlying a textured surface. This technique is commonly used in haptic rendering and generalizes well to virtual objects with textures. However, it introduces a step change in rendered force near entry points. In the current study, we performed collision detection by considering the location of the stylus tip relative to a sinusoidal textured surface as proposed by Ho, Basdogan and Srinivasan [8]. This algorithm, thereafter referred to as the "new algorithm", is computationally more expensive and may not generalize well around concave corners. It has, however, the advantage of rendering a continuously changing force magnitude near entry points. Interested readers can refer to the Appendix for more details on collision detection algorithms for textured objects.

In the rest of this paper, we report the stiffness thresholds for perceptually stable haptic texture rendering measured with the new collision detection algorithm. We also present some results on the characterization of the proximal stimuli that caused the perception of instability. In the interest of space, we will not repeat the details on experimental methods that can be found in [2] and [3].

2. Methods

2.1. Apparatus

A PHANToM force-reflecting haptic interface (model 1.0A with a stylus and encoder gimbals; nominal maximum stiffness = 3.5 N/mm) was used in all experiments to render virtual textured surfaces.

2.2. Subjects

Three subjects (one male, S1 and two females, S2 and S3) participated in the experiments. S1 and S2 also participated in our previous experiments [2]. S1 and S3 are experienced users of the PHANToM. S2 had not used any haptic interface prior to her participation in [2]. The average age of the subjects was 33.6 years old. All subjects are right-handed and reported no known sensory or motor abnormalities with their upper extremities.

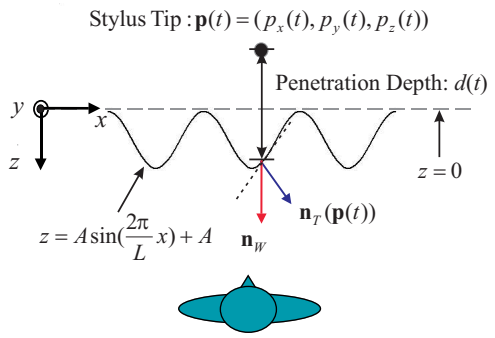


Figure 1. Illustration of parameters used in texture rendering.

2.3. Stimuli

The virtual textured surfaces were modeled as one-dimensional sinusoidal gratings with amplitude A and spatial wavelength L, superimposed on a flat surface (see Fig. 1). Sinusoidal gratings have been widely used as basic building blocks for textured surfaces in the studies on haptic texture perception [11][19] and as a basis function set for modeling real haptic textures [18].

Two texture rendering methods, one based on [12] (denoted by $\mathbf{F}_{mag}(t)$) and the other on [8] (denoted by $\mathbf{F}_{vec}(t)$), were employed. The former renders a force with a constant direction normal to the underlying flat wall ($\mathbf{n}_W$ in Fig. 1). The latter renders a force in a direction that stays normal to the sinusoidal textured surface ($\mathbf{n}_T(\mathbf{p}(t))$ in Fig. 1). See [2] for further details.

The only difference between the stimuli used in the present study and those used in [2] was the way with which penetration depth was calculated. Previously, collision detection was based on the z-coordinate of the stylus position and the underlying flat wall. Penetration depth was calculated as

$$d_1(t) = \begin{cases} 0 & \text{if } p_z(t) > 0 \\ A\sin\left(\frac{2\pi}{L}p_x(t)\right) + A - p_z(t) & \text{if } p_z(t) \le 0 \end{cases} \quad (1)$$

where $\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t))$ is the position of the stylus tip. In the present study, collision detection was performed on the boundary defined by the sinusoidal textured surface. Penetration depth was calculated as

$$d_2(t) = \begin{cases} 0 & \text{if } p_z(t) > h(p_x(t)) \\ A\sin\left(\frac{2\pi}{L}p_x(t)\right) + A - p_z(t) & \text{if } p_z(t) \le h(p_x(t)) \end{cases} \quad (2)$$

where $h(p_x(t)) = A\sin\left(\frac{2\pi}{L}p_x(t)\right) + A$ is the height of texture model at $p_x(t)$ as shown in Fig. 1.

The stimulus used in this study was uniquely defined by the amplitude (A) and wavelength (L) of the sinusoidal texture model, the surface stiffness (K), and the texture rendering method.

2.4. Experimental conditions

As in [2], two exploration modes, free exploration and stroking, were tested in order to examine the effects of user interaction patterns on instability perception. Four experiments were defined by the combinations of the two texture rendering methods and the two exploration modes (see Table 1). Five combinations of A and L were used per experiment. They were a subset of the conditions tested in [2]. The dependent variable measured was again the maximum stiffness below which the rendered textured surface was perceived to be stable.

2.5. Procedure

The experimental procedure was essentially the same as that employed in [2] with the following exceptions. The order of the four experiments, as well as that of the five experimental conditions within each experiment, was randomized for each subject. Based on preliminary testing, the maximum stiffness (K_{max}) for free exploration (Exps. I and III) was set to 1.0 N/mm. The K_{max} value used for stroking (Exps. II and IV) was 1.6 N/mm. The increment of stiffness ΔK was 0.05 N/mm for all conditions. Further details can be found in [2].

3. Results

Stiffness thresholds (denoted by K_T) under which the textured surfaces were perceived to be stable are shown in Fig. 2 for the four experiments. In each plot, the average K_T values are represented by circles. The error bars represent ± 1 standard deviations. Dashed lines are used to project the symbols to the A-L plane. The ranges of means were 0.1707–0.5497, 0.2490–0.7097, 0.0204–0.0256, and 0.3877–0.7786 N/mm for Exp. I, II, III, and IV, respectively. A five-way ANOVA performed on the pooled data of all subjects showed that all five factors of texture rendering method, exploration mode, A, L, and subject significantly affected the values of K_T ($R^2 = 0.89$).[1]

In Fig. 2, one can observe a general trend for K_T to increase with L and to decrease with A, except for the means in Exp. III that are extremely low. This trend was confirmed for the data collected in Exps. I and IV by three-way ANOVA analyses with the three factors of A, L, and subject performed on the pooled data, and by two-way ANOVA analyses with the two factors of A and L performed on individual data. In Exp. II, L was not a significant factor in the

[1] Due to the space limit, not all relevant statistics are shown in this section.

Table 1. Experimental conditions of psychophysical experiments.

Experiment	Texture Rendering Method	Exploration Mode	Texture Model Parameters (A (mm), L (mm))
I	$\mathbf{F}_{mag}(t)$	Free Exploration	(0.5, 2.0), (1.0, 1.0), (1.0, 2.0), (1.0, 4.0), (2.0, 2.0)
II	$\mathbf{F}_{mag}(t)$	Stroking	(0.5, 2.0), (1.0, 1.0), (1.0, 2.0), (1.0, 4.0), (2.0, 2.0)
III	$\mathbf{F}_{vec}(t)$	Free Exploration	(0.5, 2.0), (1.0, 1.0), (1.0, 2.0), (1.0, 4.0), (2.0, 2.0)
IV	$\mathbf{F}_{vec}(t)$	Stroking	(0.5, 2.0), (1.0, 1.0), (1.0, 2.0), (1.0, 4.0), (2.0, 2.0)

data of subjects S1 or S3, but was significant in the data of subject S2. It also turned out to be statistically significant in the pooled data, even though the three K_T means at $A = 1.0$ mm in Fig. 2(b) were almost constant over the L range (1–4 mm). Thus, we conclude that the effect of L on K_T values ($p = 0.0022$ in the pooled data) was weaker than that of A ($p < 0.0001$) in Exp. II. In Exp. III, the stiffness thresholds were uniformly close to zero within one step size (0.05 N/mm) regardless of the A or L values.

In terms of texture rendering method, the textured surfaces rendered with $\mathbf{F}_{mag}(t)$ resulted in significantly larger stiffness thresholds than those with $\mathbf{F}_{vec}(t)$ for the free exploration mode (mean difference = 0.3353, $p < 0.0001$), but not for the stroking mode. In fact, the reverse was true (average difference = -0.1138, $p < 0.0001$). This was mainly due to the relatively large differences in the data of subject S3 obtained in Exps. II and IV that exhibited a trend opposite of that observed with subjects S1 and S2. In terms of exploration mode, stroking resulted in higher stiffness thresholds than free exploration (average difference = 0.2597, $p < 0.0001$). Statistical analyses performed on individual subject's data confirmed these trends observed in the pooled data.

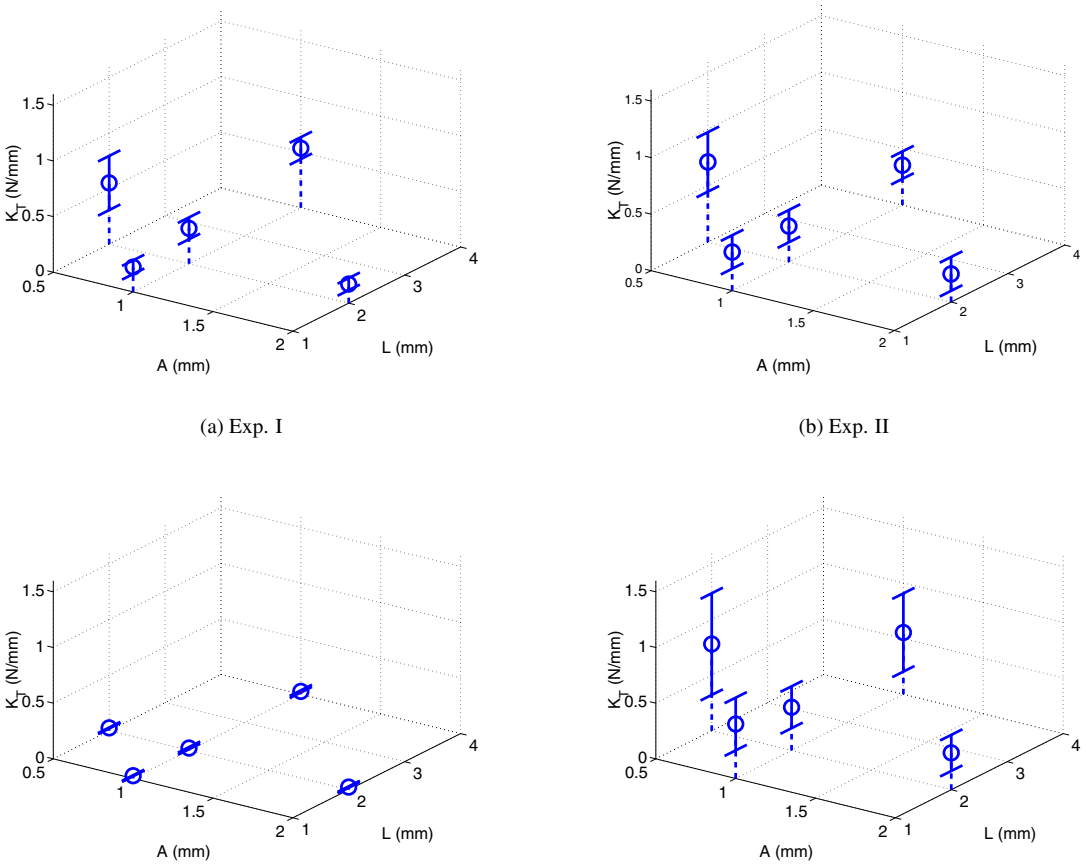
4. Discussion

In this study, we measured the maximum stiffness values under which a virtual textured surface rendered with a force-reflective device was perceptually "clean" and stable. The collision detection algorithm employed in this study was based on the relative position of the stylus tip and the height of the sinusoidal surface texture (see Eq. 2). The experimental conditions used in the current study were a subset of those used in the previous study [2] in which we employed a collision detection algorithm based on the relative position of the stylus tip and the flat wall underlying the textured surface (see Eq. 1). We can examine the effects of collision detection algorithms on the perceived stability of haptically rendered virtual textures by comparing the results from our previous and current studies.

Table 2 presents the ranges and means of stiffness thresholds K_T from the current study and from [2] for each experiment. Due to intersubject differences, only data from subjects S1 and S2 who participated in both experiments were used for the comparison. Based on the fact that the collision detection algorithm used in the current study removed a step change in force magnitude near the entry points that was present in the algorithm used in [2], we expected the stiffness thresholds obtained in the current study to be larger or at least no smaller than those obtained in [2]. Our results confirmed this expectation. Statistically, the increase in stiffness thresholds was significant in Exps. I, II, and IV (average difference = 0.2687, 0.0487 and 0.1960 N/mm, respectively; $p < 0.0001$), but not in Exp. III (average difference = 0.0026 N/mm; $p = 0.2403$). Numerically, only the threshold increases in Exps. I and IV were significant.

To gain further insight into the percepts experienced by the subjects, we studied the proximal stimuli (position, force and acceleration) using apparatus and methods described in [3]. In Exp. I ($\mathbf{F}_{mag}(t)$, free exploration) and Exp. II ($\mathbf{F}_{mag}(t)$, stroking), subjects could feel "buzzing" noise due to energy in the high frequency range when the stiffness was well above the measured threshold. For example, Fig. 3(a) shows the spectral density for position signal $p_z(t)$ (perpendicular to the wall underlying the textured surface) taken in Exp. I. Also shown for comparison is the human detection thresholds reproduced from [17] (dashed line with triangles). We can observe several spectral peaks starting at 169 Hz (marked f_{ins} for instability frequency) that are as much as 7 dB above human detection thresholds. These signals were recorded while the stylus tip was positioned near the textured surface. Accordingly, we refer to this type of instability as "entry instability." Fig. 3(b) shows a similar data recording from Exp. II. We can observe a spectral peak at around 71 Hz (marked f_{tex} for texture frequency) that was located using the estimated texture frequency from the spatial frequency L of the texture model and the measured stroking velocity (see [3] for details of the estimation method). In addition, we observed prominent spectral peaks in the high frequency region starting at around 150 Hz (marked f_{ins}) that are as much as 25 dB above human detection thresholds. Results similar to those shown in Fig. 3 were consistent with subjects' report that they judged textured surfaces as being unstable through the detection of high-frequency buzzing noises.

When the values of surface stiffness were slightly above the thresholds, no prominent spectral peaks were measured. Instead, subjects often reported that the textured surfaces



(a) Exp. I

(b) Exp. II

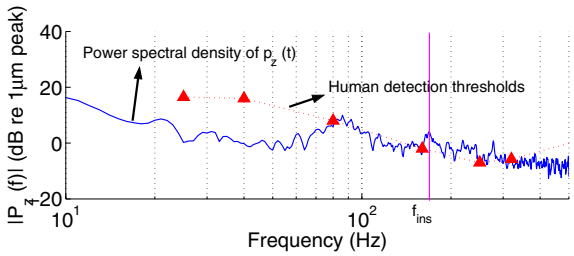
(c) Exp. III

(d) Exp. IV

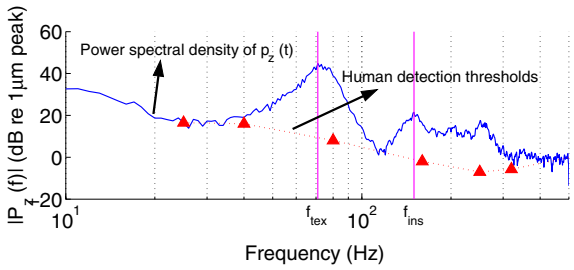
Figure 2. Results of psychophysical experiments.

Table 2. Comparison of stiffness thresholds measured in the current and previous experiments [2]. All stiffness values are in N/mm. Only data from subjects S1 and S2 are shown.

Experiments	Current experiments ($d_2(t)$)		Previous experiments ($d_1(t)$)	
	Range	Mean	Range	Mean
I. $\mathbf{F}_{mag}(t)$, free exploration	0.1813 – 0.5383	0.3486	0.0586 – 0.1023	0.0799
II. $\mathbf{F}_{mag}(t)$, stroking	0.2490 – 0.6410	0.3603	0.4488 – 0.1664	0.3116
III. $\mathbf{F}_{vec}(t)$, free exploration	0.0181 – 0.0260	0.0235	0.0097 – 0.0367	0.0209
IV. $\mathbf{F}_{vec}(t)$, stroking	0.3254 – 0.4638	0.3808	0.0718 – 0.3292	0.1848

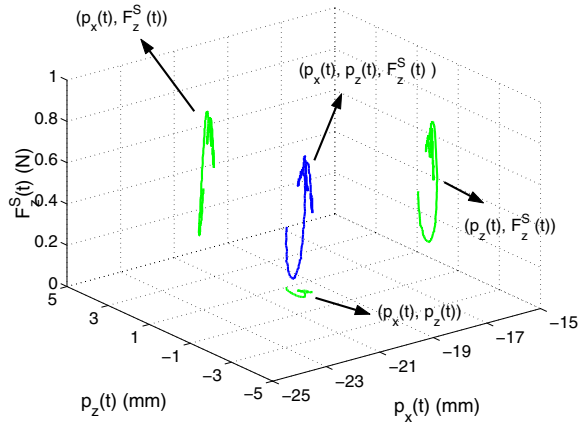


(a) Free exploration and (A, L, K) = (1 mm, 2 mm, 1.0 N/mm)

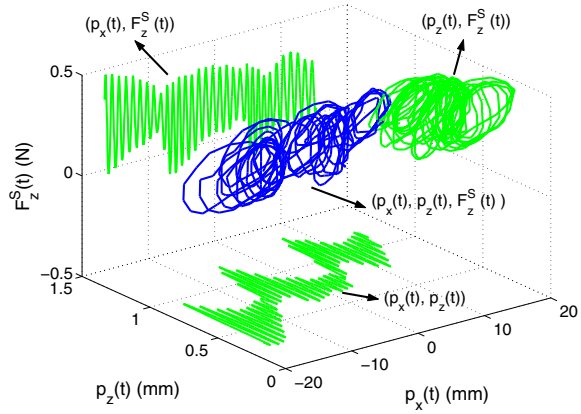


(b) Stroking and (A, L, K) = (1 mm, 2 mm, 1.2 N/mm)

Figure 3. Characteristics of buzzing when $\mathbf{F}_{mag}(t)$ was used.



(a) Free exploration and (A, L, K) = (1 mm, 1 mm, 0.4 N/mm)



(b) Stroking and (A, L, K) = (1 mm, 1 mm, 0.5 N/mm)

Figure 4. Characteristics of aliveness when $\mathbf{F}_{mag}(t)$ was used.

appeared to be "alive." We interpreted such sensations as due to a perceptible change in force while the stylus was perceived to be stationary in space. An example of such measurement for Exp. I using a stiffness value about one standard deviation above the measured threshold is shown in Fig. 4(a). Here, the measured force along the cylindrical axis of the stylus, $F_z^S(t)$, is plotted with respect to the displacement of the stylus, $(p_x(t),\ p_z(t))$, for a short period of time (400 ms). Note that $F_z^S(t)$ was not a function of $p_y(t)$ since the texture model did not vary along the y-axis. Also shown are the three projections. The projection on the $p_x(t)$-$p_z(t)$ plane shows that the tip of the stylus moved by less than 0.56 mm in x direction and 0.94 mm in z direction. This movement magnitude was probably below human detection thresholds while the hand was moving in free space. The corresponding change in force magnitude, however, was large enough to be perceptible ($\max F_z^S(t) - \min F_z^S(t) = 0.59$ N). Fig. 4(b) shows data collected during Exp. II using stroking. Again, the change in force magnitude was on the order of 0.5 N despite a short time period (400 ms). These recordings seemed to be consistent with the perception of "aliveness."

The types of instability perceived in Exps. III and IV using textures rendered with $\mathbf{F}_{vec}(t)$ were very similar to those perceived in the previous study that used collision detection algorithm $d_1(t)$ in Eq. 1. When the subjects pushed the stylus deep into the textured walls, they could feel a strong high frequency buzzing ("inside instability"). As in [3], we were able to observe spectral peaks around 220 Hz under conditions when virtual textures were perceived to be unstable. One type of instability that was unique to the $\mathbf{F}_{vec}(t)$ rendering method was that of "ridge instability." When a real stylus rests on the ridge of a real surface with sinusoidal gratings, the reaction force and friction of the surface combine to counterbalance the force exerted by the user's hand holding the stylus, thereby creating an equilibrium. The force rendered by $\mathbf{F}_{vec}(t)$, however, was determined solely based on the local texture geometry and did not take into account

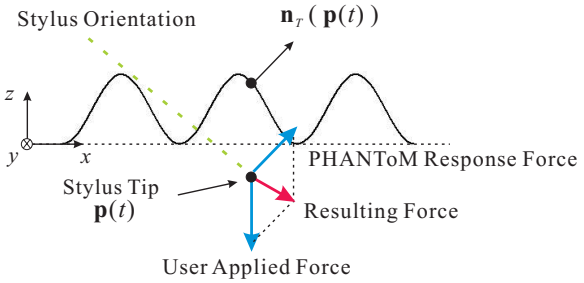


Figure 5. Ridge instability.

the direction of user applied force. This is illustrated in Fig. 5. In this figure, it is assumed that the force applied by the user was normal to the plane underneath the texture. According to the environment model, the force applied by the PHANToM was always in the direction of the surface normal $\mathbf{n}_T(\mathbf{p}(t))$. As a result, the net force exerted on the tip of the stylus (the sum of the forces applied by the user and the PHANToM) was directed towards the valley of the sinusoidal grating. Therefore, the subject who tried to rest the stylus on the ridge could feel the stylus being actively pushed into the valley.

In summary, the types of perceived instability discussed above are sometimes due to control instability (e.g., entry instability of textured surfaces rendered with $\mathbf{F}_{mag}(t)$ using high stiffness and inside instability of those rendered with $\mathbf{F}_{vec}(t)$), and sometimes due to improper environment model (e.g., ridge instability of texture surfaces rendered with $\mathbf{F}_{vec}(t)$).

Collision detection algorithms clearly influence the perceived instability of virtual surface textures, but their effects turned out to depend on other factors such as texture rendering method and exploration mode. Of particular interest is why using the new algorithm that removed discontinuities of force commands at entry points (i.e., $d_2(t)$) increased the stiffness thresholds in Exps. I and IV, but not in Exps. II and III. An examination of recorded position data $p_z(t)$ in Exp. II revealed that the stylus tip mostly remained inside the textured surface when subjects stroked across it. This explains why perceived instability in Exp. II did not depend on the use of $d_1(t)$ or $d_2(t)$. We then examined the data of $p_z(t)$ recorded in Exp. IV and found that the stylus tip was frequently lifted to the outside of the textured surface. Unlike in Exp. II that used $\mathbf{F}_{mag}(t)$, forces rendered in Exp. IV using $\mathbf{F}_{vec}(t)$ had lateral (i.e., in x direction in Fig. 1) components that sometimes opposed the stroking motion. To keep stroking against the opposing forces, subjects tended to move the stylus tip away from the textures slightly. Finally, the stiffness thresholds in Exp. III obtained with $d_1(t)$ and $d_2(t)$ were all too small (i.e., within one ΔK) to show any significant differences. These small thresholds were mainly due to the inside instability.[2]

Appendix: Collision Detection Algorithms For Textured Objects

In general, the detection of collision between an interaction tool of a haptic interface and virtual objects is a computationally complex task that has to be executed within a fraction of the haptic update interval. A collision detection problem is usually reduced to finding a point on an object surface that is closest to the position of the interaction tool. Many collision detection algorithms for efficient haptic rendering have been studied for general geometrical object models such as polygonal and NURBS models (see [10] for a review).

Collision detection becomes much more complex once textures (micro-geometry of objects) are mapped to the object surfaces. For simplicity of further discussion, we assume that the macro-geometries of the virtual objects are modeled with polygons. Difficulty in collision detection for textured objects arises from two sources. One source is the nonlinearity associated with the representation of textured object surfaces. As a result, iterative numerical algorithms are often required to determine a point on such a textured surface with a minimum distance from the interaction tool. The computation time is usually too long for such algorithms to be useful for haptic rendering. The other source is the lack of a global representation of the boundaries of the textured virtual objects. In a typical implementation, computer memory stores the polygons and the texture model separately. The texture model is locally mapped onto a point on the polygon for calculating the height and/or perturbed normal at that point. It is often infeasible to search for a global minimum using only the local information.

Few studies have explicitly considered the problem of collision detection in haptic texture rendering, except that of Ho, Basdogan, and Srinivasan [8]. In their algorithm, a polygon with a minimum distance from the tip of the interaction tool is determined first (for example, polygon L_1 in Fig. 6(a)). The distance between the polygon and the tool tip (l) is compared to the height of a texture model projected on the normal of the polygon (h) to declare an occurrence of collision. This algorithm efficiently makes correct decisions when the tool tip is not too close to the edges of polygons. It can fail when the interaction tool is in contact with a bump on another polygon, instead of a bump on top of the polygon found by the minimum distance criterion. This

[2] We should note that the original inventors of this method were aware of the problem of inside instability, and developed a heuristic algorithm that interpolates the direction of a force vector between the normal to the texture model (for small penetration depth) and that to the underlying surface (for large penetration depth). See [8] for details.

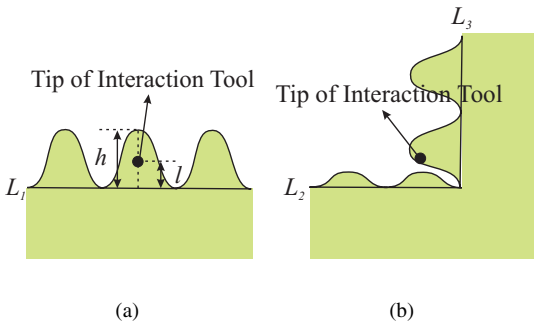


Figure 6. Examples for collision detection of textured virtual objects.

often happens when the interaction tool is near the edges of polygons, as shown in Fig. 6(b). In this case, the tool tip touches a bump on polygon L_3. The algorithm would fail to detect it since polygon L_2 is closer to the tool tip than L_3. To the best of our knowledge, no general solution has been proposed for collision detection in haptic texture rendering.

In our previous work, we started with the most basic texture rendering method that can be easily generalized to the rendering of textured objects (see $d_1(t)$ in Eq. 1), knowing that it introduced (small) discontinuity in force magnitudes. In the present study, we continued our investigation with an explicit representation of the textured wall in collision detection (see $d_2(t)$ in Eq. 2) to remove the discontinuity.

Acknowledgement

This work was supported in part by a National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award under Grant 9984991-IIS, and in part by an NSF award under Grant 0098443-IIS.

References

[1] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, June 1999.

[2] S. Choi and H. Z. Tan. An analysis of perceptual instability during haptic texture rendering. In *Proceedings of the 10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 129–136, 2002.

[3] S. Choi and H. Z. Tan. A study on the sources of perceptual instability during haptic texture rendering. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1261–1268, 2002.

[4] M. A. Costa and M. R. Cutkosky. Roughness perception of haptically displayed fractal surfaces. In *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 69-2, pp. 1073–1079, 2000.

[5] J. P. Fritz and K. E. Barner. Stochastic models for haptic texture. In *Proceedings of SPIE's International Symposium on Intelligent Systems and Advanced Manufacturing – Telemanipulator and Telepresence Technologies III*, pp. 34–44. 1996.

[6] R. B. Gillespie and M. R. Cutkosky. Stable user-specific haptic rendering of the virtual wall. In *Proceedings of the ASME International Mechanical Engineering Congress and Exhibition*, vol. 58, pp. 397–406. ASME, 1996.

[7] B. Hannaford, J.-H. Ryu, and Y. S. Kim. Stable control of haptics. In M. McLaughlin, editor, *Touch in Virtual Environments: Proceedings of USC Workshop on Haptic Interfaces*, pp. 47–70. Prentice Hall, 2001.

[8] C. Ho, C. Basdogan, and M. A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence*, 8(5):477–491, 1999.

[9] N. Hogan. Controlling impedance at the Man/Machine interface. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1626–1631, 1989.

[10] J. M. Hollerbach and D. E. Johnson. Virtual environment rendering. To appear in Human and Machine Haptics, 2000.

[11] S. J. Lederman, R. L. Klatzky, C. L. Hamilton, and G. I. Ramsay. Perceiving roughness via a rigid probe: Psychophysical effects of exploration speed and mode of touch. *Haptics-e (http://www.haptics-e.org)*, 1(1), 1999.

[12] T. H. Massie. Initial haptic explorations with the phantom: Virtual touch through point interaction. Master's thesis, MIT, Feb. 1996.

[13] B. E. Miller, E. Colgate, and R. A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6):712–719, 2000.

[14] M. D. R. Minsky. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display*. PhD thesis, MIT, June 1995.

[15] A. M. Okamura, J. T. Dennerlein, and R. D. Howe. Vibration feedback models for virtual environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 674–679, 1998.

[16] J. Siira and D. K. Pai. Haptic texturing - a stochastic approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 557–562, 1996.

[17] R. T. Verrillo. Effect of contactor area on the vibrotactile threshold. *The Journal of the Acoustical Society of America*, 35(13):1962–1966, 1963.

[18] S. A. Wall and W. S. Harwin. Modeling of surface identifying characteristics using Fourier series. In *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 67, pp. 65–71, 1999.

[19] J. M. Weisenberger, M. J. Krier, and M. A. Rinker. Judging the orientation of sinusoidal and square-wave virtual gratings presented via 2-DOF and 3-DOF haptic interfaces. *Haptics-e (http://www.haptics-e.org)*, 1(4), 2000.