vol. 24, issue 3, pp. 527-536 (2008)

On the Synthesis of Haptic Textures

Gianni Campion *Student Member, IEEE* and Vincent Hayward *Fellow, IEEE*

***Abstract*— Advanced, synthetic haptic virtual environments require textured virtual surfaces. We found that texturing smooth surfaces often reduces the system passivity margin of a haptic simulation. As a result, a smooth virtual surface that can be rendered in a passive manner may loose this property once textured. We propose that any texture algorithm is associated with a characteristic number that expresses the relative change in loop gain. We further found that a passive virtual interaction can have severe unwanted artifacts if the synthesized force field is not conservative. The energy characteristics of seven algorithms are analyzed. Finally a new texture synthesis algorithm, which operates by modulating a friction force during scanning, is shown to have several advantages over previous ones.**

***Index Terms*— Haptic Interfaces, Haptic textures, Passivity, Force reflective virtual environments, High-fidelity haptic simulation, Mechatronics.**

I. Introduction

THE "virtual wall" is considered to be a benchmark problem in haptic simulation [1], [2], [3], [4], [5]. Nevertheless, advanced, high-fidelity haptic simulations demand to consider more general cases than smooth, low curvature virtual object boundaries that virtual walls can represent.

This article considers the question of augmenting the simulation of rigid or deformable objects with surface texture. The aim is to provide a method to systematically analyze the properties of a given texture simulation algorithm, since many algorithms have been proposed [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], and many are yet to be designed.

An approach to designing haptic synthesis algorithms for simulating basic mechanical interactions between objects is to consider algorithms that provide fundamental properties shared with the interactions that these algorithms are supposed to simulate. Those properties should include general physical properties such as mathematical continuity of the force response and conservation of energy. It could be further desired to provide resemblance of the synthetic force responses with actual physical responses [17]. Passivity is also an important property of synthetic environments because, once provided, the simulation results no longer depend on the dynamic properties of the hand interacting with it [18].

In this paper, it is found that texture synthesis algorithms frequently increase the *effective stiffness* of an originally non-textured object. This algorithm-specific gain, if not accounted for, can lead to the loss of passivity during a virtual haptic interaction. A characteristic number is introduced to express the relative increase of stiffness caused by a given algorithm when texture is added to a smooth virtual surface. In addition, the cause for unintended artifacts, such as low frequency oscillations [19], [20], observed when using certain algorithms is elucidated and attributed to the creation of non-conservative virtual force fields. A noteworthy consequence is that even if a synthesis method yields a passive interaction, it can produce simulation artifacts that severely impair the realism of a simulation by causing virtual objects to feel active [21].

II. Assumptions

A. Parametrization

Most texturing algorithms, see Section IV, assume the existence of a bounded height function h defined on a smooth surface. For analysis purposes it is convenient to consider that each scan path defines a curve parametrized by its arc length, $\chi(\eta)$, which is located on the smooth surface, Fig 1a. This path also defines a moving Frenet frame located at a point on the surface parametrized in (u, v). This frame provides local coordinates $\xi(u, v)$, $\nu(u, v)$, and $\zeta(u, v)$, the tangent, normal, and binormal vectors respectively. A height function, $h(u, v)$, can then be naturally defined by the distance measured between the textured surface and the smooth boundary along the normal $\nu(u, v)$.

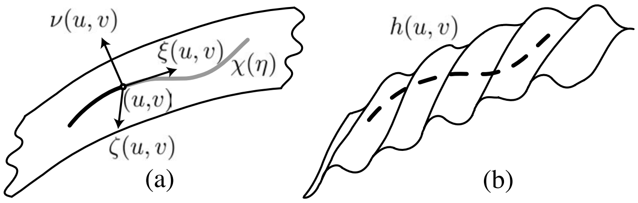


Fig. 1. Parametrization of a tool interaction with a textured surface.

Intuitively, if a surface has low curvature, the effects of texture dominate. We therefore restrict the present study to surfaces with zero curvature and straight paths. Specializing the analysis to the cross plane x-z, then suffice to study the case of $z = h(x, 0)$ when h is defined on the surface $z = 0$ and when the scan is along x. With suitable changes of coordinates, it is possible to extend the present analysis to account for the effects of surfaces with high curvature or for the effects of blending functions along the edges of polygonal meshes. These extensions are left to future work.

The continuity and differentiability requirements of h are discussed later in this article in greater detail, but in general we assume that while $h(x, y)$ can be periodic, stochastic, or otherwise, it must be band-limited so the Nyquist-Shannon sampling theorem can apply. We also consider that the haptic device has a limited bandwidth, which can be assumed to be determined by its first mechanical resonance F_0. This is a good assumption because it is difficult to robustly compensate beyond the first mode in closed loop [22].

G. Campion and V. Hayward are with the Haptics Laboratory, Center For Intelligent Machines, McGill University, Montréal, Québec, H3A 2A7 Canada. (e-mail: champ@cim.mcgill.ca and hayward@cim.mcgill.ca)

B. Limits

A set of necessary, yet not sufficient conditions must be met for a haptic texture to be properly synthesized [23]. Given the assumptions just stated, the analysis can be restricted to the case of a sinusoidal texture with spatial period l and amplitude A. Calling $\mathcal{T}$ the system sampling period, v the scanning velocity of the manipulandum, δ the spatial resolution of the device, b the force resolution of the device, A the maximum amplitude of the force to be displayed, and A_0 the maximum stiffness achievable by the device, there are six inequalities which must be met:

1. Scanning velocity limit	$\alpha\, v\, \mathcal{T}/l < 1$
2. Low speed reconstruction limit	$\beta\, \delta/l < 1$
3. High speed reconstruction limit	$\alpha\, \delta/l < 1$
4. Force reconstruction limit	$\gamma\, b/l < A$
5. Gain limit	$2\,\pi\, A/l < A_0$
6. Device structural limit	$v/l < F_0$

In the above, α, β, and γ are the numbers of samples required in a period for a correct causal reconstruction (typically 5 to 10). In this paper, we focus on the fifth inequality and assume that the five others are met. The notation is this paper is by and large consistent with that in [24] and [23], yet here, stiffnesses are denoted by 'κ' to avoid conflict with singular values noted σ.

III. Control Analysis

A. Control Passivity Condition

We first derive a corollary of Theorem 1 in reference [18], which is a generalization of Colgate and Schenkel's passivity condition [25]. Our corollary states that the largest singular value of the Jacobian matrix of the force field describing the synthetic force response must be smaller that the smallest singular value of the damping matrix of the device divided by twice the sampling period,

Corollary 1: Let $\boldsymbol{s}(\eta) = [x(\eta), y(\eta), z(\eta)]^\top$ be a trajectory of the virtual tool, which is in general different from $\chi(\eta)$. A texturing algorithm generates a force trajectory in the static field $\boldsymbol{f}(x, y, z)$. Call $\mathbf{B}$ the device damping matrix with smallest singular value $\sigma_0(\mathbf{B})$, and $\mathbf{J}_{\boldsymbol{f}}$ the Jacobian matrix of $\boldsymbol{f}(x, y, z)$ with largest singular value $\sigma_n(\mathbf{J}_{\boldsymbol{f}})$. Given time steps i

$$\forall i,\ \|\mathbf{J}_{\boldsymbol{f},i}\|_2 = \sigma_n(\mathbf{J}_{\boldsymbol{f},i}) \le \frac{\sigma_0(\mathbf{B})}{2\mathcal{T}}, \tag{1}$$

is a sufficient condition for the passivity of the virtual environment.

Proof: Adapt Condition 1 of Theorem 1 of [18]:

$$|\boldsymbol{f}(\boldsymbol{s}_i) - \boldsymbol{f}(\boldsymbol{s}_{i-1})| \le \frac{\sigma_0(\mathbf{B})}{2\mathcal{T}} |\boldsymbol{s}_i - \boldsymbol{s}_{i-1}|. \tag{2}$$

To a first order approximation, if $\mathbf{J}_{\boldsymbol{f}}$ is the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{s})$, then

$$|\mathbf{J}_{\boldsymbol{f},i}\ [\boldsymbol{s}_i - \boldsymbol{s}_{i-1}]| \le \|\mathbf{J}_{\boldsymbol{f},i}\|_2 |\boldsymbol{s}_i - \boldsymbol{s}_{i-1}|, \tag{3}$$

therefore if $\|\mathbf{J}_{\boldsymbol{f},i}\| \le \sigma_0(B)/2\mathcal{T}$ then

$$|\mathbf{J}_{\boldsymbol{f},i}\ [\boldsymbol{s}_i - \boldsymbol{s}_{i-1}]| \le \frac{\sigma_0(\mathbf{B})}{2\mathcal{T}} |\boldsymbol{s}_i - \boldsymbol{s}_{i-1}|. \tag{4}$$

which proves the Corollary by applying the triangular inequality to (4) and (3). ■

In short, this Corollary means that the algorithm used to synthesize texture has an effect on the passivity of the virtual object. For memoryless algorithms, the effect can be quantified independently of the path chosen to explore the surface. For algorithms with memory, a worse-case analysis can be performed. It is a generalization of Condition 5 of Section II.

B. Characteristic Number of Algorithms

Under the assumption that the vector field $\boldsymbol{f}$ is linear in the stiffness of the underlying smooth wall κ_0, then we define

$$q = \frac{\sigma_n(\mathbf{J}_{\boldsymbol{f}})}{\kappa_0} \tag{5}$$

to be a characteristic number associated with the algorithm generating $\boldsymbol{f}$. This number expresses a stiffness increase when simulating a "texturized" boundary. Suppose for example that a device is capable of rendering a smooth virtual wall of stiffness κ_0. Once textured, in order to preserve passivity the original wall stiffness should be reduced to $\kappa_1 = \kappa_0/q$. This notion applies to any texturing algorithm studied in this article. The determination of q requires finding the maximum singular value of $\mathbf{J}_{\boldsymbol{f}}$ inside the regions where device is intended to produce textures.

The knowledge of the characteristic numbers is useful to design virtual environments. For example, it will enable a designer to predict that doubling the spatial frequency of a texture requires dividing the nominal stiffness by two. It must be noted however, that it cannot be used as a tool to directly compare algorithms since its value can depend on different sets of parameters.

C. Conservativity and Passivity In Virtual Environments

Physically, a system is said to be conservative if the work done to modify it is equal to the change of its internal energy. Equivalently, if work is done to modify the system through *any* cycle, the system is conservative if, and only if, this work is zero. In classical mechanics, barring friction and internal dissipation, all forces are conservative [26]. Conservativity is therefore a desirable property of any virtual environment intended to replicate objects which are not actuated, as originally indicated by Salisbury et al. [21]. Later, we will use two equivalent properties to study the algorithms: the gradient of a scalar function is a conservative vector field and if a vector field has zero curl then it is conservative.

Passivity, in turn, is typically defined with reference to input-output properties in a system [27]. In this view, a system is seen as a black box and dissipativity is defined in an abstract sense in terms of an input-output product (e.g. forces and displacements for mechanical systems). A system must always extract energy through time to be passive. If we combine a conservative system with dissipative elements, such as friction or dampers, seen from outside the box, the system will necessarily be passive. Nevertheless, if we combine a generative system with elements that are sufficiently dissipative, the system, as a whole, may remain passive.

vol. 24, issue 3, pp. 527-536 (2008)

For the design of *multi-dimensional* virtual mechanical environments [18], conservativity is therefore a more desirable property than input-output passivity since one could construct systems that behave passively for some trajectories but have severely non-physical responses for others. A useful consequence of enforcing conservativity for a virtual environment is to neatly decouple the performance of a synthesis algorithm from that of a device and from its control. Three cases can arise:

1) Condition (1) does not hold, then unwanted energy may be injected in the system.
2) Condition (1) holds and the algorithm is generative, then the simulation may inject energy for some trajectories even if the simulation is passive for other trajectories.
3) Condition (1) holds and the virtual environment is conservative, then for any trajectory the system will always extract energy from the user.

Unless the simulation is intended to be generative, the third case is the only desirable case.

IV. Analysis of Algorithms

We now investigate seven different algorithms for their energy properties in x-z coordinates only (See Section II). Fig. 2 and Table I summarize the algorithms, all sharing a nominal stiffness κ_0.

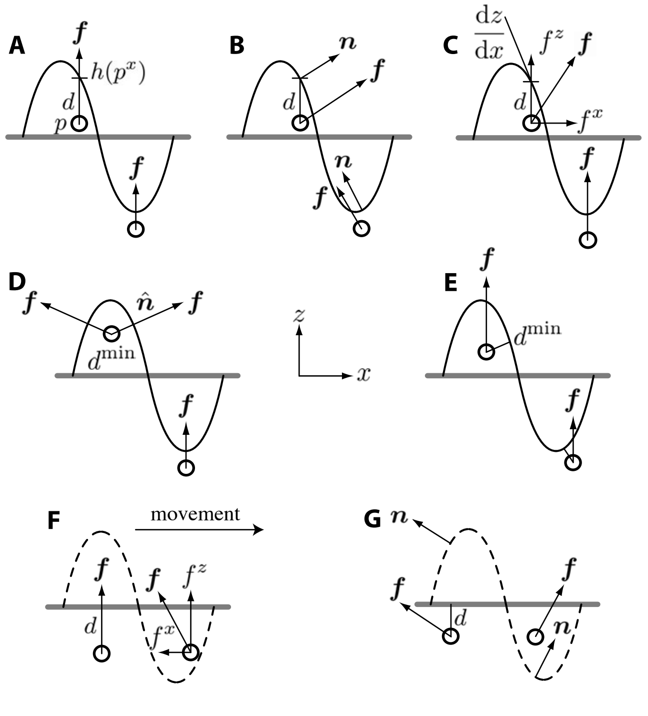


Fig. 2. Different texture synthesis algorithms. Circles represent the position of the handle in virtual space. For each algorithm we show the force in two different locations. Thick gray lines symbolize the non-textured surface. Solid lines indicate the virtual object boundary that is used in the penetration computation. Dashed lines represent the textured surfaces when the penetration is computed from the nominal surface. **A**, **B** are the algorithms used in [19]. **C** is discussed in [16]. **E** and **D** are derived from the 'god object' approach [28]. **F** is a dry-friction-modulation based algorithm introduced here. **G** is from [29].

Algorithms can be classified into three groups according to how penetration is computed.

TABLE I

Expressions for $\boldsymbol{f}(\boldsymbol{s})$. See Fig. 2 for the corresponding diagrams and main text for details.

A	$[0, -\kappa_0 d]^\top$ [19]	**B** & **G**	$-\kappa_0 d\,\boldsymbol{n}$ [19] & [29]
D	$-\kappa_0 d^{\min}\,\hat{\boldsymbol{n}}$ [28]	**E**	$[0, -\kappa_0 d^{\min}]^\top$
F	$[-\kappa(x)d, -\kappa_0 d]^\top$	**C**	$[-\kappa_0 \dfrac{\partial h}{\partial p^x}\, d, -\kappa_0 d]^\top$ [16]

- Algorithms **A**, **B**, and **C** determine d by projection $d = d^z = p^z - h(p^x)$, where p is the position of the interaction point in virtual space.
- Algorithms **E** and **D** use the minimum distance between p and the textured surface $d = d^{\min}$ as a measure of penetration.
- Algorithms **F** and **G** compute d as the distance to the nominal surface, not to the textured surface: $d = p^z$.

Remark 1: It is important to recognize that an accurate reproduction of haptic texture is a difficult problem, not only from the point of the view of the device performance limitations [30], but also from the view point of the physics involved. There is a variety of complex micro-mechanical phenomena related to the tribological properties of the materials in contact and to the mutual geometrical relationships of the tool and the surface. It is a common experience that the texture of rough paper is greatly affected by the instrument used to write on it. Compare a pencil and a ball-pen. For the same paper, the micro-mechanics and the feel are different. In one case, the interaction depends on graphite dry friction and on the other, on a ball rolling on viscous fluid bed. Moreover, a given texture synthesized with a given algorithm often produces different perceptual experiences when felt through haptic devices of different types. In view of these complexities, the algorithms considered in this paper are neither discussed from the view point of their physical relevance nor from that of their perceptual value—all have plausible physical interpretations and all may give interesting results. They are discussed strictly for their energy properties.

Remark 2: Calculations and experiments concern textures having one single amplitude and single spatial frequency. However, under the assumption that the haptic device used to synthesize these textures is reasonably linear under small-signal operating conditions, then these results extend to any texture made of a sum of sinusoids. More generally, if the textures are not periodic, then similar calculations can be done by assuming that the rate of change of h is limited, i.e., it is Lipschitz with bound M. Then, the quantity $2\pi A/l$ would be replaced by M.

Remark 3: The characteristic number of algorithms **B**, **G**, **F**, **C** depends on the penetration inside the virtual wall because the normal force and lateral force components are coupled. The simulation may lose passivity if penetration exceeds a limit. A simple approach to solve this problem is to limit the value of the penetration d^z to an acceptable limit $d^z_{\max}$ and use this value to compute the lateral force component.

vol. 24, issue 3, pp. 527-536 (2008)

A. Grooved boundary—force normal to surface [19] (**A**)

1) Field: Algorithm **A** considers a virtual-wall-like force component that is normal to the nominal non-textured surface. There is no force component along the direction tangent to the nominal virtual surface. The texture sensation then arises by virtue of oscillations of the system finger-haptic device caused by a vector field aligned along one single direction.

$$\boldsymbol{f}_{\mathbf{A}}(\boldsymbol{s}) = \begin{cases} [0, -\kappa_0\, d^z]^\top & \text{if } d^z < 0, \\ [0, 0]^\top & \text{otherwise.} \end{cases} \tag{6}$$

The penetration is computed from the boundary of the texture along the z direction.

2) Jacobian: Inside the boundary, when $d < 0$, the Jacobian matrix of $\boldsymbol{f}_{\mathbf{A}}(\boldsymbol{s})$ works out to be:

$$\mathbf{J}_{\boldsymbol{f}_{\mathbf{A}}}(\boldsymbol{s}) = -\kappa_0 \begin{bmatrix} 0 & 0 \\ -h'(p^x) & 1 \end{bmatrix}. \tag{7}$$

3) Characteristic number: The norm of $\mathbf{J}_{\boldsymbol{f}_{\mathbf{A}}}(\boldsymbol{s})$ is

$$\|\mathbf{J}_{\boldsymbol{f}_{\mathbf{A}}}(\boldsymbol{s})\|_2 = \kappa_0 \sqrt{1 + [h'(p^x)]^2} \tag{8}$$

giving

$$q_{\mathbf{A}} = \kappa_{\mathbf{A}}^{\max} / \kappa_0 = \sqrt{1 + [h'(p^x)]^2} \tag{9}$$

that expresses the increase in stiffness incurred by simulating a virtual textured with Algorithm **A**.

For a sinusoidal surface $h(d^x) = A \sin(2\pi p^x / l)$ the maximum value of the norm of the Jacobian matrix is:

$$\kappa_{\mathbf{A}}^{\max} = \kappa_0 \sqrt{1 + [2\pi A / l]^2} \tag{10}$$

when $p^x = 0 \mod (\pi / l)$. Since typically, $2\pi A / l \gg 1$, the gain is nearly proportional to κ_0, A, and $1/l$. Thus, it has a form similar to that of the Condition 5 of Section II.

4) Conservativity: By inspection of (7) we can conclude that the curl is not zero, hence, the algorithm does not generate a conservative field. For example, with $h(p^x) = A \sin(2\pi p^x / l)$, when traversing the closed path (0,0)→(l/4,0)→(l/4,A)→(0,A)→(0,0), see Fig. 3, the virtual environment will generates energy $\Delta E = +1/2 \kappa_0 A^2$.

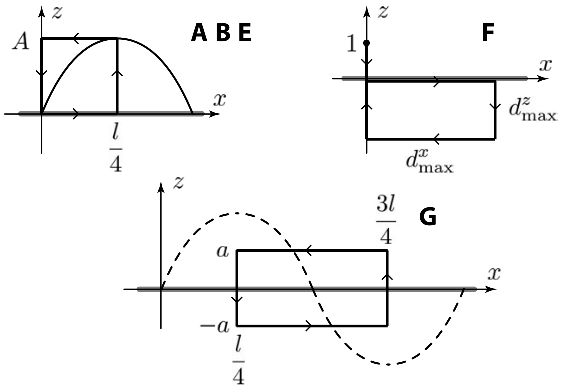


Fig. 3. The energy balance along these paths is positive showing that the corresponding force fields are not conservative.

B. Grooved boundary—force normal to groove [19] (**B**)

1) Field: Algorithm **B** is similar to the previous one, but computes a force that is aligned with the normal to the virtual surface being explored. The point where the normal is evaluated is given directly by p^x. The field is

$$\boldsymbol{f}_{\mathbf{B}}(\boldsymbol{s}) = \begin{cases} -\kappa_0\, d^z \dfrac{[-h'(p^x), 1]^\top}{\sqrt{1 + h'(p^x)^2}} & \text{if } d^z < 0, \\ [0, 0]^\top & \text{otherwise.} \end{cases} \tag{11}$$

2) Jacobian and characteristic number: A manageable closed form could not be found but can be numerically evaluated.

3) Conservativity: When $h(p^x) = A \sin(2\pi p^x / l)$, cycling through the path (0,0)→($\frac{l}{4}$,0)→($\frac{l}{4}$,A)→(0,A)→(0,0) generates

$$\Delta E = \frac{\kappa_0\, A^2}{2} + \frac{\kappa_0\, L^2}{4\, \pi^2} \left(1 - \sqrt{\frac{L^2 + 4\, A^2\, \pi^2}{L^2}} \right) \tag{12}$$

Remark 4: Algorithms **A** and **B** do not generate conservative force fields and are known to cause a feel of activity.

C. Change of Height [16], [11] (**C**)

Algorithm **C**, described in [16], is the extension of that used in the Sandpaper System to three dimensions [6]. The force field does not follow the normal of the texture; it depends on the position but not on the trajectory of the handle. A related algorithm, presented in [11], computes a force with a tangential component that is proportional to the rate of change of the height of the virtual textured surface with respect to the curvilinear abscissa of the scan path. The algorithm results from the observation that the intensity of an interaction force should relate both to the shape of the surface and to the manner in which it is explored. It was observed that this algorithm can be viewed as a way to eliminate non-working forces [31]. In this paper we refer to the static version of this algorithm [16].

1) Field: Given a height map $h(p^x)$, a potential $U_{\mathbf{C}}$ function of the normal deflection can be defined:

$$U_{\mathbf{C}} = \begin{cases} -\kappa_0 [d^z]^2 / 2 & \text{if } d^z < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The gradient of $U_{\mathbf{C}}$ gives the force field

$$\boldsymbol{f}_{\mathbf{C}}(\boldsymbol{s}) = \begin{cases} -\kappa_0 [-d^z h'(p^x), d^z]^\top & \text{if } d^z < 0, \\ [0, 0]^\top & \text{otherwise.} \end{cases} \tag{14}$$

which extends the rendering described in [11] to account for normal deflection.

2) Jacobian: The Jacobian matrix of this field

$$\mathbf{J}_{\boldsymbol{f}_{\mathbf{C}}}(\boldsymbol{s}) = -\kappa_0 \begin{bmatrix} [h'(p^x)]^2 - d^z h''(p^x) & -h'(p^x) \\ -h'(p^x) & 1 \end{bmatrix} \tag{15}$$

depends on the second derivative of the height field h. This fact suggests that the texture stiffness grows with the *curvature* of the simulated surface. This actually has a nice physical interpretation since we would expect the interaction forces, and hence the gain of the synthesis, to become very large in the sharp asperities and crevices of an irregular surface.

vol. 24, issue 3, pp. 527-536 (2008)

3) Characteristic number: It must be numerically evaluated.

4) Conservativity: Given by construction.

Remark 5: A key difference between algorithms **B** and **C** is the normalization factor $\sqrt{1+(h'(p^x)^2)}$ in (11). This factor prevents algorithm **B** from generating a conservative force field whereas algorithm **C**'s field is conservative.

D. Variant 1 derived from the 'god-object' method (**D**)

Algorithm **D** is a first variant of the 'god-object' method [28], which at all times minimizes the amount of penetration inside a virtual surface.

1) Field: If p_h is the surface point closest to p, given a texture with a generating function $p^z_h = h(p^x_h)$, a potential U_{D} can be defined

$$U_{\mathsf{D}} = \begin{cases} -\kappa_0[(p^x - p^x_h)^2 + (p^z - p^z_h)^2]/2 & \text{inside,} \\ 0 & \text{outside.} \end{cases} \tag{16}$$

where (p^x_h, p^z_h) are the coordinates of p_h.

By taking the gradient, we obtain

$$\boldsymbol{f}_{\mathsf{D}}(\boldsymbol{s}) = \begin{cases} -\kappa_0[(p^x - p^x_h), (p^z - p^z_h)]^\top & \text{inside,} \\ 0 & \text{outside.} \end{cases} \tag{17}$$

By virtue of being a gradient, the algorithm is conservative.

2) Jacobian: The Jacobian matrix inside the boundary is (see Appendix II)

$$\mathbf{J}_{\boldsymbol{f}_{\mathsf{D}}}(\boldsymbol{s}) = -\frac{\kappa_0}{(1+h'(p^x_h)^2)} \begin{bmatrix} h'(p^x_h)^2 & -h'(p^x_h) \\ -h'(p^x_h) & 1 \end{bmatrix}. \tag{18}$$

This matrix is singular. Moving in a direction orthogonal to the minimum penetration vector keeps the force constant as long as the potential is differentiable.

3) Characteristic number: Notice that $\|\mathbf{J}_{\boldsymbol{f}_{\mathsf{D}}}(\boldsymbol{s})\|_2 = \kappa_0$, showing no dependency on the boundary function. In that, its differs from algorithms **A** and **C**. The characteristic number is 1, a nice property indeed.

4) Conservativity: Given by construction.

Remark 6: Algorithm **D** has the peculiarity that it synthesizes a discontinuous force field, see Fig. 2, thus causing undesirable artifacts. For a sinusoidal texture, the discontinuities occur at the maxima of the sinusoidal function. For a general profile, the artifacts arise when the probe is equidistant from two or more different point of the boundary. The corresponding potential is continuous but not differentiable, even when the boundary is smooth, which is not physical unless shocks are acceptable.

E. Variant 2 derived from the 'god-object' method (**E**)

Algorithm **E** is a second variant of the 'god object' method.

1) Field: Here, the force field is such that the synthesized force is always normal to surface, Fig. 2(d). The force field is

$$\boldsymbol{f}_{\mathsf{E}}(\boldsymbol{s}) = \begin{cases} \left[0, -\kappa_0\sqrt{(p^x - p^x_h)^2 + (p^z - p^z_h)^2}\right]^\top & \text{inside,} \\ [0, 0]^\top & \text{outside.} \end{cases} \tag{19}$$

2) Jacobian: Inside the boundary, the Jacobian matrix is

$$\mathbf{J}_{\boldsymbol{f}_{\mathsf{E}}}(\boldsymbol{s}) = -\frac{\kappa_0 \begin{bmatrix} 0 & 0 \\ (p^x - p^x_h) & (p^z - p^z_h) \end{bmatrix}}{\sqrt{(p^x - p^x_h)^2 + (p^z - p^z_h)^2}}. \tag{20}$$

3) Characteristic number: The norm of the matrix is

$$\|\mathbf{J}_{\boldsymbol{f}_{\mathsf{E}}}(\boldsymbol{s})\|_2 = \kappa_0 \tag{21}$$

and the characteristic number is also 1.

4) Conservativity: Unfortunately, the form of the Jacobian matrix shows that the force field is non conservative. In fact, on the closed path (Fig. 3) (0,0)→($\frac{l}{4}$,0)→($\frac{l}{4}$,A)→(0,A)→(0,0) the energy change is on the segment $\frac{l}{4}$, 0→$\frac{l}{4}$,A only where force and displacement are aligned. Hence $\Delta E > 0$.

Remark 7: This second variant of the 'god-object' method, algorithm **E**, while being non-conservative, does not suffer from the discontinuity problem of algorithm **D**.

F. Flat wall with modulated lateral friction (**F**)

Texture rendering with dry friction is not new, see [7]. Here we introduce formulation based on a time-free friction model (22). Instead of adding several contact force components, the tangential friction force is directly modulated by the height field of the texture.

1) Field: A tangential friction force component along x is modulated by a function of the net tangential displacement. The virtual friction is computed using the technique described in [32]. In the formula of Table I, $\kappa(.)$ has the dimension of a stiffness. Specifically, $\kappa(x) = \mu\,\kappa_0[1 - h(x)]$, where μ is Amontons' coefficient of friction, and the quantity in bracket is the modulation function. The lateral force component is combined with the normal response of an ordinary wall. This results in a synthesis that is independently tunable in the two directions, i.e. a "DC" nominal normal component and a varying, oscillatory lateral component. The force field is

$$\boldsymbol{f}_{\mathsf{F}}(\boldsymbol{s}) = \begin{cases} -\kappa_0\left[\mu[1-h(p^x)]\dfrac{d^x}{d^x_{\max}}p^z, p^z\right]^\top, & p^z < 0, \\ [0,0]^\top \text{otherwise.} \end{cases} \tag{22}$$

where $d^x_{\max}$ is the maximum pre-sliding tangential deflection [32]. For this algorithm we assume additionally that $0 \le h(p^x) \le 1$.

2) Jacobian: The Jacobian matrix is:

$$\mathbf{J}_{\boldsymbol{f}_{\mathsf{F}}}(\boldsymbol{s}) = -\kappa_0 \begin{bmatrix} \mu\dfrac{p^z}{d^x_{\max}}\left(\dfrac{\mathrm{d}d^x}{\mathrm{d}p^x} - h(p^x)\dfrac{\mathrm{d}d^x}{\mathrm{d}p^x} - h'(p^x)\right) & \mu[1-h(p^x)]\dfrac{d^x}{d^x_{\max}} \\ 0 & 1 \end{bmatrix}. \tag{23}$$

This algorithm has memory in the term $\mathrm{d}d^x/\mathrm{d}p^x$. The worst case for the condition number of the Jacobian needs to be investigated. According to [32]:

$$\text{stick phase:} \begin{cases} d^x < d^x_{\max} \\ \dfrac{\mathrm{d}d^x}{\mathrm{d}p^x} = 1 \end{cases}, \text{slip phase:} \begin{cases} d^x = d^x_{\max} \\ \dfrac{\mathrm{d}d^x}{\mathrm{d}p^x} = 0 \end{cases}. \tag{24}$$

vol. 24, issue 3, pp. 527-536 (2008)

For the example of $h(p^x) = A\sin(2\pi p^x/l)$ we can maximize each entry of the Jacobian matrix.

$$\mathbf{J}_{\boldsymbol{f}_{\mathsf{F}:\text{slip}}}(\boldsymbol{s}) = -\kappa_0 \begin{bmatrix} 2\pi\mu p^z A/l & 2\mu \\ 0 & 1 \end{bmatrix} \tag{25}$$

and

$$\mathbf{J}_{\boldsymbol{f}_{\mathsf{F}:\text{stick}}}(\boldsymbol{s}) = -\kappa_0 \begin{bmatrix} 2\mu p^z(1/d^x_{\max} + \pi A/l) & 2\mu \\ 0 & 1 \end{bmatrix} \tag{26}$$

are the maximum values of the Jacobian matrix during the slip and stick phases respectively.

3) Characteristic number: The symbolic expression for the norm of $\mathbf{J}_{\boldsymbol{f}_{\mathsf{F}:\text{stick}}}$ is cumbersome but the Appendix I can be consulted for the expression of $q_{\mathsf{F}} = \|\mathbf{J}_{\boldsymbol{f}_{\mathsf{F}}}\|_2/\kappa_0$.

Notice that in (26) the gain of the lateral force component depends on p^z. Because of the user input, p^z can grow unbounded, giving in principle an arbitrarily large gain. To address this problem, p^z must be clamped to a maximum $d^z_{\max}$. We then have a three-way trade off between A, l and $d^z_{\max}$. Note that realistic values for $d^x_{\max}$ as well as for $d^z_{\max}$ may range from 10^{-2} m (biological tissues) to 10^{-6} m (metals).

4) Conservativity: The Jacobian matrix (23) shows that the algorithm does not in general produce a conservative field. The sliding friction is guaranteed to be dissipative but energy can be generated during the stick phase. For $A = 0$ and the path $(0,1)\to(0,-a)\to(d^x_{\max},-a)\to(d^x_{\max},-d^z_{\max})\to(0,-d^z_{\max})\to(0,1)$, (Fig. 3), the energy gain is

$$\Delta E = +\frac{\kappa_0\,\mu}{2} d^z_{\max} d^x_{\max} \tag{27}$$

The algorithm is generative only in the stick phase which is at most $2d^x_{\max}$ wide. This confined energy gain does not create artifacts in the following experiments. However, this gain might be a problem when rendering highly compliant deformable bodies ($d^x_{\max} \approx 1$ cm) because the non conservative energy is proportional to $d^x_{\max}$.

G. Force Shading (**G**)

Lastly, the 'force-shading' algorithm **G**, which synthesizes a force normal to the texture, was implemented [29].

1) Field: The field is

$$\boldsymbol{f}_{\mathsf{G}}(\boldsymbol{s}) = \begin{cases} -\kappa_0\, p^z \dfrac{[-h'(p^x), 1]^\top}{\sqrt{1+(h'(p^x))^2}}\,\boldsymbol{n} & \text{if}\, p^z < 0, \\ [0,0]^\top & \text{otherwise.} \end{cases} \tag{28}$$

2) Jacobian: The Jacobian matrix does not easily afford a closed-form solution.

3) Characteristic number: While an expression could not be found for the algorithm's characteristic number, it was possible to tune it empirically.

4) Conservativity: The algorithm is non-conservative. On the path $(l/4,a)\to(l/4,-a)\to(3\,l/4,-a)\to(3\,l/4,a)\to(l/4,a)$ for example, (Fig. 3), the energy gain is

$$\Delta E = -1/2\kappa_0 a^2 + \Delta E_2 + 1/2\kappa_0 a^2 + 0 = \Delta E_2 > 0. \tag{29}$$

since force and the displacement are aligned only on the segment $(l/4,-a)\to(3\,l/4,-a)$.

TABLE II
SUMMARY OF ALGORITHMS PROPERTIES.

	Char. number	Conservativity	Continuity
A	$\sqrt{1+[2\pi A/l]^2}$	No	If h continuous
B	No closed-form	No	If h' continuous
C	No closed-form	Yes	If h' continuous
D	1	Yes	No. Even if h continuous
E	1	No	If h continuous
F	Appendix I	Almost	If h continuous
G	No closed-form	No	If h' continuous

H. Summary

Table II summarize the discussion of the previous sections.

V. EXPERIMENTAL VALIDATION

In a first step, we use an elastic rubber band to verify that, indeed, the mere fact of adding texture to an otherwise smooth virtual surface does increase its apparent stiffness and may lead to a loss of passivity.

In a second step, an operator actively explores textured surfaces. The energy supplied to the haptic device is recorded. The reason behind the second set of tests is that lack of conservativity of the underlying force field, can only be revealed through active scanning of the simulated surface. It can be verified that, as predicted, different algorithms have radically different energetic behaviors when tested under similar conditions.

To perform these tests under the best conditions possible, we use a two degrees-of-freedom device, which was rigorously characterized [33]. From these tests, we established that its spatial resolution measured by external backdriving is better than 15 μm. The sampling period is $\mathcal{T} = 10^{-4}$ s and the highest texture spatial period is $l = 1$ mm. Since the scanning velocity is smaller than 0.2 m^{-1}, the number of samples α and β is always larger than 50 and the maximum frequency of 200 Hz is well within the 400 Hz-wide 3 dB flatband of the device [33]. For the experiments, we use the variant in [34], which enables us to program high quality viscous damping in the machine, See Fig. 4.

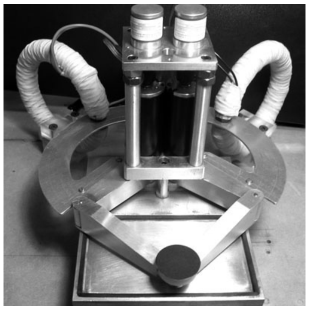


Fig. 4. The 2D device used in the experiments. It has two direct-drive motors and two programmable eddy-current brakes. An elastic band (not shown) is attached to the handle during the passivity experiments.

vol. 24, issue 3, pp. 527-536 (2008)

A. Passivity Experiments

A virtual wall, algorithm **W**, was set up with the force field

$$\boldsymbol{f}_{\mathsf{W}}(\boldsymbol{s}) = \begin{cases} [0, -\kappa_0\, p^z]^\top & \text{if}\, p^z < 0, \\ [0, 0]^\top & \text{otherwise.} \end{cases} \tag{30}$$

The intrinsic physical dissipation in the device is an imperceptible amount of friction in the joints of device, internal friction in the rubber band, and air drag around the linkages. We also add viscous damping when the handle is inside the virtual object, 4.7 mN·m·s in each joint. Next, the manipulandum is thrust into the wall such that a lack of passivity is indicated by the onset of spontaneous activity. Referring to Fig. 5, stiffness is tuned so that the smooth virtual wall is passive. When exactly the same wall is textured with algorithm **A** or **F**, stable limit cycles occur. The conditions are $\kappa_0 = 2\,000$ N·m^{-1}, $l = 1$ mm, $A = 0.8$ mm, $\mu = 0.8$, $d^x_{\max} = d^z_{\max} = 1$ mm. We then apply the theory of Section III to shows that if we account for the characteristic numbers of algorithms, $q_{\mathsf{F}} = 8.3$ and $q_{\mathsf{A}} = 5.1$, we can ensure passivity.

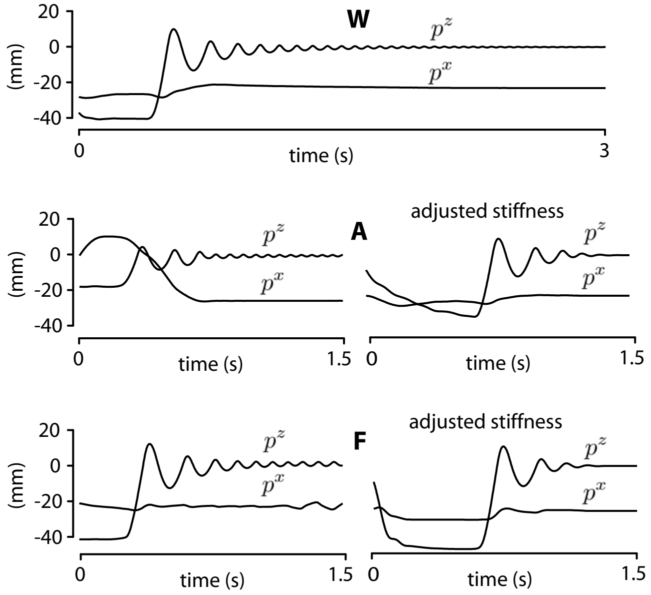


Fig. 5. Plots were obtained by thrusting the manipulandum with an elastic band onto virtual walls. For the wall, **W**, algorithm **A** (left), and algorithm **F** (left), when $\kappa_0 = 2\,000$ N·m^{-1}. The wall is passive but once textured, the system enters a limit cycle. Behavior of algorithm **A**, when κ_0 is reduced by the corresponding characteristic number (right). Same for algorithm **F**.

B. Conservativity Experiments

A subject explores a virtual wall textured with selected algorithms discussed earlier and their energetic behavior is recorded. In principle, to test conservativity, the system should follow closed paths. Admittedly, it is rather hard to ask a subject to do this accurately. In practice, we can replace cycling through a closed trajectory by oscillations around a nominal value and monitor the *average* work made during many cycles. Leaving everything else unchanged, we can compare algorithms by computing $\sum_0^N f_x \Delta p^x$ and $\sum_0^N f_z \Delta p^z$ where Δp^x and Δp^y are the incremental displacements of the position observed during one sample period, that is, every 100 μs in the present experiments. The quantities f_z and f_x are the force commands sent to the device.

A textured wall is set up with parameters $\kappa_0 = 1\,000$ N·m^{-1}, $l = 2$ mm, $A = 0.8$ mm, $\mu = 0.8$, $d^x_{\max} = d^z_{\max} = 1$ mm. The analog dampers are energized when the handle is inside the virtual body to 4.0 mN·m·s. The passivity margin of this experimental condition is larger than the one shown for the passivity tests. Following the usual convention, a decreasing energy indicates that the energy is supplied by the virtual environment. Fig. 6 shows the energy balance during a single stroke along the texture that lasts approximately 1 second.

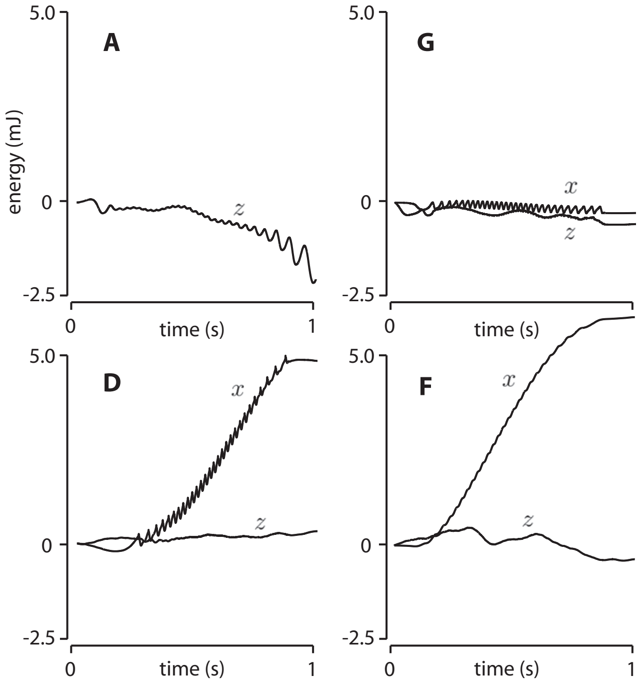


Fig. 6. Plots obtained by computing the energy supplied to the device dragging the handle along a textured wall. The user moved the handle while attempting to keep the penetration as constant as possible.

With algorithm **A** there is a noticeable energy gain in the z direction indicative of a generative behavior during a single continuous lateral motion. There is no energy associated with the lateral direction. By comparison, the 'force-shading' algorithm **G** is marginally passive in the normal as well as in the lateral directions although it is not strictly conservative. Algorithm **D**, employing the 'god-object' approach, is marginally passive in the normal direction z and is clearly passive in the lateral direction x for an equivalent textural vibration. This was despite the fact that algorithm **D** synthesizes a discontinuous force field that causes the hills to feel 'clipped' and distorted. Algorithm **F** is marginally passive in the normal direction z and is passive in the lateral direction x as a result of the dissipative nature of the friction model.

C. Surface Activity

Choi and Tan referred to low frequency vibrations experienced by users when exploring surfaces synthesized by

vol. 24, issue 3, pp. 527-536 (2008)

algorithms **A** and **B** [19]. This artifact is likely to be associated to the fact that the force fields generated by these algorithms are not conservative. In our experience, this sensation arises even if the virtual surface is made passive by reducing its nominal stiffness sufficiently, or even by adding physical damping.

To confirm this hypothesis, we can test algorithms **A**, **B**, **E**, and **D** with an 'enlarged bump' created by setting $\kappa_0 = 1\,000$ N·m^{-1}, $A = 20$ mm and $l = 20$ mm in order to magnify the phenomenon; a damping coefficient of 4.0 mN·m·s is added inside the virtual wall. All of these algorithms feel active with the exception of algorithm **D** which feels passive, yet discontinuous. The results of employing algorithm **A** and **D** were recorded in order to view the 'magnified' dynamics of the interaction. Earlier, it was seen that for algorithm **A** the characteristic number is proportional to A/l while the energy gain on an horizontal path is proportional to A^2. This means that it is possible to find a generative texture algorithm that has passivity characteristics. The results are in Fig. 7.

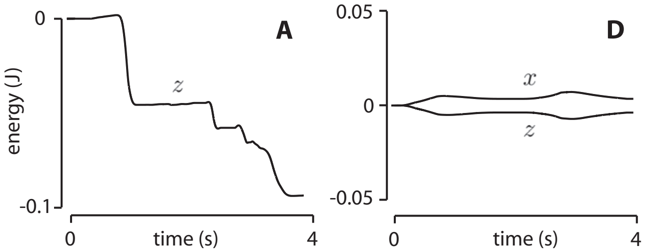


Fig. 7. The microdynamics of interaction with a texture boundary were magnified by exaggerating the size of the feature. Here, a 20 mm bump was explored and two synthesis algorithms were compared.

Algorithm **A**, because it 'pushes' the handle in a direction that differs from its movements, makes it difficult for the user to interact with the virtual object. Contact is often lost as a result of the sharp energy increase created at the contact with the surface. The energy plot, Fig. 7, shows these energy jumps. Algorithm **D**, however, maintains an overall energy balance. From these tests, we surmise that algorithms which yield non-conservative fields feel 'alive' although, locally, the complete system may be passive, recall Section III-C. In particular, algorithm **A** is passive when the probes moves perpendicularly to the surface as shown in the experiment with the elastic band.

VI. CONCLUSION

The properties of common texture synthesis algorithms were analyzed in terms of their effective stiffness, that is, of the incremental stiffness they cause. Texturing a surface generally decreases the passivity margin of a given virtual mechanical environment. We found that any specific algorithm has a characteristic number that expresses this gain succinctly. In some cases this number is constant or even equal to 1. In others cases, it depends on the texture parameters and is most frequently of the form $\propto A/l$, where A is the height of the texture and l its spatial period.

It is confirmed that absence of surface activity and passivity are related but distinct notions. Surface activity is related to unphysical artifacts generated by algorithms synthesizing non-conservative force fields. Passivity is related to the system dynamics and can always be ensured by reducing the closed loop gain, by increasing the sample rate, or by adding dissipative elements. Conservativity, on the hand, is an intrinsic property of algorithmic synthesis which may or may not be ensured, independently of any particular hardware or control technique.

We demonstrated that some widely used texturing algorithms do create non-conservative force fields for typical exploration paths, linking energy gain to artifacts which are not intended by the application programmer. We showed that continuous and conservative texture force fields can be created using a change-of-height approach. The 'god-object' approach was found to be nominally conservative. It leaves the effective stiffness equal to the original stiffness but produces undesirable discontinuities. A variant that creates a force field that is always normal to the original surface cured this problem but caused the algorithm produce non-conservative force fields and hence active surfaces. We introduced a new synthesis algorithm which creates textures by modulating the lateral friction force when scanning a surface. It provides parameters that can be tuned to deliver realistic sensations and a simple formulation for its characteristic number. This algorithm is not particularly advantageous in terms of its effective stiffness but the dissipative nature of the friction field makes the texture feel realistic and passive.

Finally, the perceptual qualities of the various algorithms remain to be investigated with regard to new haptic device interface concepts such as that in [35], [34].

VII. ACKNOWLEDGMENTS

The authors would like to thank Mohsen Mahvash and Andrew H. C. Gosline for insightful comments on earlier drafts of this paper. This research was supported in part by the Institute for Robotics and Intelligent Systems, and NSERC, the Natural Sciences and Engineering Research Council of Canada.

APPENDIX I
CHARACTERISTIC NUMBER OF ALGORITHM F

An upper bound for q_{F} $h(p^x) = A\sin(2\pi p^x/l)$ is

$$
\begin{aligned}
q_{\mathsf{F}} &\leq \left\| \begin{matrix} 2\mu p^z(1/d^x_{\max} + \pi A/l) & 2\mu \\ 0 & 1 \end{matrix} \right\|_2 \\
&\leq \frac{\sqrt{2}}{2\,d^x_{\max}} \Big[4(\mu p^z)^2 + 8(\mu p^z)^2\,\pi A d^x_{\max}/l \\
&\quad +4(\mu p^z \pi A d^x_{\max}/l)^2 + 4(\mu d^x_{\max})^2 + {d^x_{\max}}^2 \\
&\quad +\big(16(\mu^4 p^z)^4 + 32\mu^2(\mu^2 d^x_{\max} p^z)^2 \\
&\quad +64(\mu p^z)^4 \pi A d^x_{\max}/l + {d^x_{\max}}^4 + 96(\mu p^z)^4(\pi A d^x_{\max}/l)^2 \\
&\quad +64(\mu p^z)^4(\pi A d^x_{\max}/l)^3 + 64\mu^4 {d^x_{\max}}^3 {p^z}^2 \pi A/l \\
&\quad -16{d^x_{\max}}^3(\mu p^z)^2 \pi A/l + 16(\mu p^z \pi A d^x_{\max}/l)^4 \\
&\quad +32(\mu^4 d^x_{\max})^4 (p^z \pi A/l)^2 - 8\,{d^x_{\max}}^4(\mu p^z \pi A/l)^2 \\
&\quad +16(\mu d^x_{\max})^4 + 8\,\mu^2\,{d^x_{\max}}^4 - 8(\mu p^z d^x_{\max})^2\big)^{1/2}\Big]^{1/2} \quad (31)
\end{aligned}
$$

vol. 24, issue 3, pp. 527-536 (2008)

Appendix II
Jacobian matrix of algorithm $\mathbf{D}$

For the god-object method, the assumptions are:

- the boundary curve $h(p^x)$ is smooth and differentiable.
- there is just one point (p^x_h, p^z_h) on h that minimizes the distance between the (p^x, p^z) and the boundary.
- probe is 'inside' the texture.

We know that:

$$(p^z - p^z_h) = -(p^x - p^x_h)/h'(p^x_h). \tag{32}$$

while the linearization around (p^x, p^z) gives:

$$\frac{\partial p^z_h}{\partial p^x} = h'(p^x_h)\frac{\partial p^x_h}{\partial p^x}. \tag{33}$$

$$\frac{\partial p^x_h}{\partial p^z} = \frac{\partial p^z_h}{\partial p^z}/h'(p^x_h). \tag{34}$$

The energy function can be described by

$$E_{\mathbf{D}} = -\kappa_0[(p^x - p^x_h)^2 + (p^z - p^z_h)^2]/2. \tag{35}$$

Differentiating (35) gives

$$\begin{aligned}
\boldsymbol{f}_{\mathbf{D}}(\boldsymbol{s})_x &= \frac{\partial E_{\mathbf{D}}}{\partial p^x} && (36)\\
&= -\kappa_0\left((p^x - p^x_h)(1 - \frac{\partial p^x_h}{\partial p^x}) - (p^z - p^z_h)\frac{\partial p^z_h}{\partial p^x}\right) &&\\
&= -\kappa_0(p^x - p^x_h) + \kappa_0\left(\frac{\partial p^x_h}{\partial p^x} - \frac{\partial p^z_h}{\partial p^x}/h'(p^x_h)\right) && (37)\\
&= -\kappa_0(p^x - p^x_h). && (38)
\end{aligned}$$

where we used equation (32), (33), and (34). The derivation of the force is complete by defining $\boldsymbol{f}_{\mathbf{D}}(\boldsymbol{s})_x = 0$ if $h'(p^x_h) = 0$.

The same reasoning can be used to derive

$$\boldsymbol{f}_{\mathbf{D}}(\boldsymbol{s})_z = -\kappa_0(p^z - p^z_h) \tag{39}$$

and $\boldsymbol{f}_{\mathbf{D}}(\boldsymbol{s})_z = 0$ if $1/h'(p^x_h) = 0$. The Jacobian can be easily computed:

$$\begin{aligned}
\mathbf{J}_{\boldsymbol{f}_{\mathbf{D}}}(\boldsymbol{s}) &= -\kappa_0\begin{bmatrix} 1 - \frac{\partial p^x_h}{\partial p^x} & -\frac{\partial p^x_h}{\partial p^z} \\ -\frac{\partial p^z_h}{\partial p^x} & 1 - \frac{\partial p^z_h}{\partial p^z}\end{bmatrix} && (40)\\
&= -\kappa_0\begin{bmatrix} 1 - \frac{\partial p^x_h}{\partial p^x} & -\frac{\partial p^z_h}{\partial p^z}/h'(p^x_h) \\ -h'(p^x_h)\frac{\partial p^x_h}{\partial p^x} & 1 - \frac{\partial p^z_h}{\partial p^z}\end{bmatrix}. && (41)
\end{aligned}$$

Knowing that:

$$\frac{\partial p^x_h}{\partial p^x} = \cos(\mathrm{atan}(h'(p^x_h)))^2 = 1/(1 + h'(p^x_h)^2) \tag{42}$$

and

$$\frac{\partial p^z_h}{\partial p^z} = \sin(\mathrm{atan}(h'(p^x_h)))^2 = h'(p^x_h)^2/(1 + h'(p^x_h)^2) \tag{43}$$

we can write

$$\mathbf{J}_{\boldsymbol{f}_{\mathbf{D}}}(\boldsymbol{s}) = -\frac{\kappa_0}{1 + h'(p^x_h)^2}\begin{bmatrix} h'(p^x_h)^2 & -h'(p^x_h) \\ -h'(p^x_h) & 1\end{bmatrix}. \tag{44}$$

References

[1] J. E. Colgate, P. E. Grafing, M. C. Stanley, and G. Schenkel, "Implementation of stiff virtual walls in force-reflecting interfaces," in *Virtual Reality Annual International Symposium*, 1993, pp. 202–208.

[2] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 465–474, 1999.

[3] J. K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24–32, 2004.

[4] J. J. Abbott and A. M. Okamura, "Effects of position quantization and sampling rate on virtual wall passivity," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 952–964, 2005.

[5] N. Diolaiti, G. Niemeyer, F. Barbagli, and J. K. Salisbury, "Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 256–268, 2006.

[6] M. Minsky and S. J. Lederman, "Simulated haptic textures: Roughness," in *Proc. ASME IMECE Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, DSC-Vol. 58*, 1996, pp. 421–426.

[7] J. Siira and D. Pai, "Haptic textures – a stochastic approach," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1996, pp. 557–562.

[8] J. P. Fritz and K. E. Barner, "Stochastic models for haptic textures," in *Proc. SPIE Vol. 2901, p. 34-44, Telemanipulator and Telepresence Technologies III*, M. R. Stein, Ed., 1996, pp. 34–44.

[9] M. Costa and M. Cutkosky, "Roughness perception of haptically displayed fractal surfaces," in *Proceedings ASME IMECE Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, vol. 69-2, 2000, pp. 1073–1079.

[10] J. M. Weisenberger, M. J. Kreier, 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*, vol. 1, no. 4, pp. –, 2000.

[11] V. Hayward and D. Yi, "Change of height: An approach to the haptic display of shape and texture without surface normal," in *Experimental Robotics VIII*, ser. Springer Tracts in Advanced Robotics, B. Siciliano and P. Dario, Eds. Heidelberg: Springer Verlag, 2003, pp. 570–579.

[12] M. A. Otaduy and M. C. Lin, "A perceptually-inspired force model for haptic texture rendering," in *Proc. 1st Symposium on Applied perception in graphics and visualization*. ACM Press, 2004, pp. 123–126.

[13] A. Crossan, J. Williamson, and R. Murray-Smith, "Haptic granular synthesis: Targeting, visualisation and texturing," in *Proceedings of the International Symposium on Non-visual & Multimodal Visualization*. IEEE press, 2004, pp. 527–532.

[14] M. A. Otaduy, N. Jain, A. Sud, and M. C. Lin, "Haptic display of interaction between textured models," in *Proceedings of IEEE Visualization*, 2004, pp. 297–304.

[15] N. Melder and W. S. Harwin, "Force shading and bump mapping using the friction cone algorithm," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. WHC 2005.*, 2005, pp. 573–575.

[16] 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, pp. 153–161, 1998.

[17] M. Mahvash and V. Hayward, "High fidelity haptic synthesis of contact with deformable bodies," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 48–55, 2004.

[18] ——, "High fidelity passive force reflecting virtual environments," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 38–46, 2005.

[19] S. Choi and H. Z. Tan, "Perceived instability of virtual haptic texture. I. Experimental studies," *Presence: Teleoperators & Virtual Environments*, vol. 13, no. 4, pp. 395–415, 2004.

[20] ——, "Perceived instability of virtual haptic texture. II. Effect of collision-detection algorithm," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 4, pp. 463–81, 2005.

[21] K. J. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic rendering : Programming touch interaction with virtual objects," in *Proceedings Symposium on Interactive 3D Graphics*. New York, NY, USA: ACM Press, 1995, pp. 123–130.

[22] K. Zhou and J. C. Doyle, *Essentials of Robust Control.* Prentice Hall, 1997.

[23] G. Campion and V. Hayward, "Fundamental limits in the rendering of virtual haptic textures," in *Proceedings of the First Joint Eurohaptics*

vol. 24, issue 3, pp. 527-536 (2008)

Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2005, 2005, pp. 263–270.

[24] V. Hayward, “Haptic synthesis,” in *Proceedings of the 8th International IFAC Symposium on Robot Control, SYROCO 2006*, 2006, pp. 19–24.

[25] J. E. Colgate and G. Schenkel, “Passivity of a class of sampled-data systems: Application to haptic interfaces,” in *Proceedings of the American Control Conference*, 1994, pp. 3236–3240.

[26] H. Goldstein, *Classical mechanics*. Addison-Wesley, 1950.

[27] D. Hill and P. Moylan, “Dissipative dynamical systems: Basic input-output and state properties,” *Journal of the Franklin Institute*, vol. 309, no. 5, pp. 327–357, 1980.

[28] C. B. Zilles and J. K. Salisbury, “A constraint-based god object method for haptic display,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS’95*, vol. 3, 1995, pp. 146–151.

[29] H. B. Morgenbesser and M. A. Srinivasan, “Force shading for haptic shape perception,” in *Proceedings of the Fifth Symposium on Haptic Interfaces for Virtual Environments and Teleoperators, ASME Dynamic Systems and Control Division*, vol. DSC 58, 1996, pp. 407–412.

[30] V. Hayward and O. R. Astley, “Performance measures for haptic interfaces,” in *Robotics Research: The 7th International Symposium*, G. Giralt and G. Hirzinger, Eds. Heidelberg: Springer Verlag, 1996, pp. 195–207.

[31] A. Frisoli, “Personal communication.” 2004.

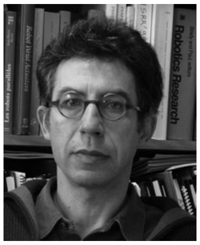
[32] V. Hayward and B. Armstrong, “A new computational model of friction applied to haptic rendering,” in *Experimental Robotics VI*, ser. Lecture Notes in Control and Information Sciences, P. Corke and J. Trevelyan, Eds., vol. 250, 2000, pp. 403–412.

[33] G. Campion, Q. Wang, and V. Hayward, “The Pantograph Mk-II: A haptic instrument,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS’05*, 2005, pp. 723–728.

[34] A. H. Gosline, G. Campion, and V. Hayward, “On the use of eddy current brakes as tunable, fast turn-on viscous dampers for haptic rendering,” in *Proceedings of Eurohaptics*, 2006, pp. 229–234.

[35] S. A. Wall and W. S. Harwin, “Effects of physical bandwidth on perception of virtual gratings,” in *Proceedings of the Symposium on Haptic Interfaces for Virtual Environments and Teleoperators, ASME Dynamic Systems and Control Division*, 2000, pp. 1033–1039.

Gianni Campion received his M.Sc. in Computer Science from University of Verona (Italy) in 2000. He joined the Haptics Laboratory at McGill University in 2002, where he is pursuing a Ph.D. in Electrical Engineering. His research interests include force feedback displays, surgical robotics, and haptic textures.



Vincent Hayward (M84SM04-FIEEE’08), Ing. Ecole Centrale de Nantes 1978; Ph.D. Computer Science 1981, University of Paris; Visiting Assistant Professor, Purdue University (1982);Chargé de Recherches at CNRS, France (1983-86), Professeur invité, Université Pierre et Marie Curie (2006); is now Professor of Electrical and Computer Engineering at McGill University. Hayward is interested in haptic device design and applications, perception, and robotics. He is leading the Haptics Laboratory at McGill University and was the Director of the McGill Center for Intelligent Machines (2001-2004). He is a co-Founder of the Experimental Robotics Symposia, Program Vice-Chair 1998 IEEE Conference on Robotics and Automation, Program Vice-Chair ISR2000, past Associate Editor IEEE Transactions on Robotics and Automation, Governing board Haptics-e, Editorial board of the ACM Transaction on Applied Perception of the IEEE Transactions on Haptics. Hayward received several best paper awards and research awards including the NASA Space Act Tech Brief Award (1991) and the E. (Ben) & Mary Hochhausen Award for Research in Adaptive Technology For Blind and Visually Impaired Persons (2002).