

# Incorporating Geometric Algorithms in Impedance- and Admittance-Type Haptic Rendering

Ryo Kikuuwe and Hideo Fujimoto  
Nagoya Institute of Technology, Nagoya, 466-8555, Japan

## Abstract

*Geometric (proxy-based) haptic rendering algorithms are widely used in impedance-type haptic rendering. Such methods are useful for developing complex virtual environment because they reduce mechanistic problems of determining reaction forces into geometric problems of updating proxy positions. This paper aims to apply geometric algorithms to wider variations of haptic rendering systems. First, we present a method to include viscous damping in impedance-type haptic rendering. Second, we propose a method to utilize geometric algorithms in admittance-type haptic rendering. The results of implementation experiments are presented.*

## 1. Introduction

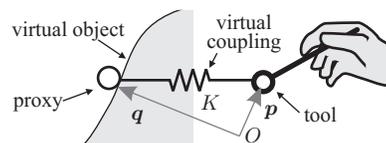
The control schemes of haptic devices can be broadly classified into two types: the impedance type and admittance type. In impedance-type haptic rendering, the actuator force is determined according to the position of the tool (the grasping portion of a haptic device) measured with position sensors such as optical encoders. This scheme is suited for sufficiently back-drivable and lightweight haptic devices. On the contrary, in admittance-type haptic rendering, the tool position is controlled to follow a desired position that is determined according to the force measured with force sensors. This control scheme is used with devices having a large inertia or high-friction joints (e.g., [1, 2, 3]) because such dynamics can be suppressed by stiff position controllers.

In both of impedance- and admittance-type haptic rendering, it is generally difficult to construct consistent algorithms to treat geometrically complex virtual objects. In the impedance type, it is convenient to use a special virtual object (a *proxy*) connected to the tool through a virtual spring-like element (a *virtual coupling*). The proxy moves together with the tool in the virtual environment, but it remains on the surface of a virtual object when the tool comes inside the virtual object. Moreover, friction and textures of

virtual surfaces can be displayed by appropriately limiting the moving speed of the proxy. In this scheme, the mechanistic<sup>1</sup> problem of determining the reaction force can be replaced by a geometric problem of updating the proxy position. It is convenient for developers (programmers) of virtual environments because the algorithms can be developed without considering physical forces. It however includes at least two problems. One is that it is unclear how to include damping in the virtual coupling in impedance-type haptic rendering. The other is that an admittance-type haptic rendering algorithm accepts a force input although a geometric algorithm requires a position input.

This paper provides solutions for the above two problems. First, we propose an impedance-type haptic rendering scheme in which a geometric algorithm is combined with viscous damping. Second, we propose an admittance-type haptic rendering scheme incorporating a geometric algorithm. The use of the proposed methods allows the use of common geometric algorithms both in impedance- and admittance-type haptic rendering.

In what follows, section 2 reviews the geometric approach that have been used in impedance-type haptic rendering. Its problems and limitations are also discussed. Section 3 presents a mechanistic interpretation on geometric algorithms and derives new techniques. Section 4 describes experimental results and Section 5 provides concluding remarks.



**Figure 1. Impedance-type haptic rendering using proxy.**

<sup>1</sup>We use the adjective ‘mechanistic’ to mean ‘of mechanics.’ Mechanics includes dynamics and statics, which concern physical forces.

## 2. Geometric Haptic Rendering Algorithms

### 2.1. Proxy-based Impedance-type Haptic Rendering

The concept of proxy was first proposed by Zilles and Salisbury [4]. They used a point-like virtual object to represent the haptic device (tool) in the virtual environment, and termed it as a *god object*. Ruspini et al. [5] extended this concept to display surface deformations, textures, and friction. They termed the virtual object as a *proxy*.

The following discussion assumes that the proxy is a point-like element and the virtual environment is an  $n$ -dimensional Cartesian space. Most of recent haptic rendering algorithms using proxies can be described in the following general form:

$$\mathbf{q}(k) = \mathcal{G}(K, \mathbf{p}(k), \mathbf{q}(k-1)) \quad (1a)$$

$$\mathbf{f}(k) = K(\mathbf{q}(k) - \mathbf{p}(k)) \quad (1b)$$

where  $k$  is a discrete-time index,  $\mathbf{p} \in \mathbb{R}^n$  and  $\mathbf{q} \in \mathbb{R}^n$  are the positions of the tool and the proxy, respectively, and  $K > 0$  is the stiffness of the virtual coupling. The function  $\mathcal{G} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a geometric algorithm that accepts the current tool position  $\mathbf{p}(k)$  and the proxy position  $\mathbf{q}(k-1)$  of the one timestep ago as inputs, and provides a new proxy position  $\mathbf{q}(k)$  as the output. The function  $\mathcal{G}$  takes the stiffness  $K$  as an argument so that its computation may depend on  $K$ . The actuator force  $\mathbf{f}(k) \in \mathbb{R}^n$  of the haptic device is determined by the positional relation between the tool and the proxy as indicated in (1b).

As a simple example, let us consider a unilateral constraint imposed by a flat object surface including the origin. Then, the function  $\mathcal{G}$  can be written as follows:

$$\mathcal{G}(K, \mathbf{p}, \mathbf{q}) = \begin{cases} \mathbf{p} & \text{if } \mathbf{n}^T \mathbf{p} > 0 \\ \mathbf{p} - \mathbf{n} \mathbf{n}^T \mathbf{p} & \text{if } \mathbf{n}^T \mathbf{p} \leq 0 \end{cases} \quad (2)$$

where  $\mathbf{n} \in \mathbb{R}^n$  ( $\|\mathbf{n}\| = 1$ ) is the outward normal direction of the surface. Fig 2 illustrates the relation among  $\mathbf{p}$ ,  $\mathbf{q}$ , and  $\mathbf{q}_{\text{new}} = \mathcal{G}(K, \mathbf{p}, \mathbf{q})$  under this definition. The new proxy position  $\mathbf{q}_{\text{new}}$  coincides with the tool position  $\mathbf{p}$  as long as the tool is outside the object (i.e.,  $\mathbf{n}^T \mathbf{p} > 0$ ). When  $\mathbf{p}$  is inside the object,  $\mathbf{q}_{\text{new}}$  is the orthogonal projection of  $\mathbf{p}$  onto the surface, and thus the proxy remains on the surface.

The following definition of the function  $\mathcal{G}$  is another example, which represents an isotropic Coulomb friction law:

$$\mathcal{G}(K, \mathbf{p}, \mathbf{q}) = \begin{cases} \mathbf{q} & \text{if } \|\mathbf{p} - \mathbf{q}\| \leq F/K \\ \mathbf{p} - \frac{F}{K} \frac{\mathbf{p} - \mathbf{q}}{\|\mathbf{p} - \mathbf{q}\|} & \text{if } \|\mathbf{p} - \mathbf{q}\| > F/K. \end{cases} \quad (3)$$

Here,  $F > 0$  represents the kinetic and maximum static friction force. Fig 3 illustrates this definition, with which

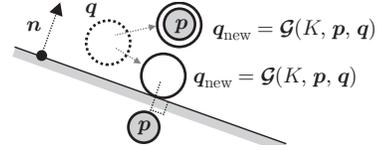


Figure 2. Unilateral constraint by a frictionless flat surface.

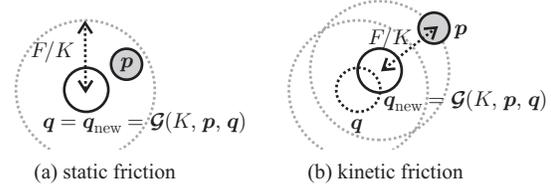


Figure 3. Isotropic Coulomb friction.

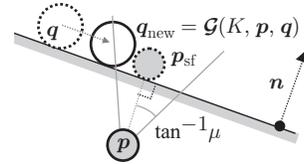


Figure 4. Unilateral constraint by a frictional flat surface.

the proxy does not move as long as the distance between the tool and the proxy is smaller than  $F/K$ , and otherwise the proxy follows the tool maintaining the distance  $F/K$ . This way of modeling Coulomb friction can be found in the literature [6, 7].

The following definition represents a flat surface that includes the origin and has the friction coefficient of  $\mu > 0$ :

$$\mathcal{G}(K, \mathbf{p}, \mathbf{q}) = \begin{cases} \mathbf{p} & \text{if } \mathbf{n}^T \mathbf{p} > 0 \\ \mathbf{q}_{\text{sf}} & \text{if } \mathbf{n}^T \mathbf{p} \leq 0 \wedge \|\mathbf{p}_{\text{sf}} - \mathbf{q}_{\text{sf}}\| \leq -\mu \mathbf{n}^T \mathbf{p} \\ \mathbf{p}_{\text{sf}} + \mu \mathbf{n}^T \mathbf{p} \frac{\mathbf{p}_{\text{sf}} - \mathbf{q}_{\text{sf}}}{\|\mathbf{p}_{\text{sf}} - \mathbf{q}_{\text{sf}}\|} & \text{if } \mathbf{n}^T \mathbf{p} \leq 0 \wedge \|\mathbf{p}_{\text{sf}} - \mathbf{q}_{\text{sf}}\| > -\mu \mathbf{n}^T \mathbf{p} \end{cases} \quad (4)$$

where  $\mathbf{p}_{\text{sf}} = \mathbf{p} - \mathbf{n} \mathbf{n}^T \mathbf{p}$ ,  $\mathbf{q}_{\text{sf}} = \mathbf{q} - \mathbf{n} \mathbf{n}^T \mathbf{q}$ , and  $\mathbf{n}$  is the outward normal vector of the surface. Fig. 4 illustrates this definition, with which the proxy position is updated so that it remains in the friction cone.

The algorithm  $\mathcal{G}$  can be more complex if the virtual environment is more complex. Saving the computational cost while maintaining the consistency of haptic rendering is the major interest of recent studies [8, 9, 10]. It nevertheless is a common procedure to determine the current proxy position  $\mathbf{q}(k)$  and the reaction force  $\mathbf{f}(k)$  as described in (1). It is convenient for developers in that a mechanistic problem of determining reaction forces can be treated with geometric consideration.

## 2.2. Limitation of Geometric Algorithms

The algorithm (1) produces an elastic force that does not include damping. It is usually preferred to include some viscous damping with a feedback of the measured velocity to reduce a bouncy feeling or to prevent the system from going unstable. It, however, is not straightforward to consistently combine the geometrically determined forces and viscous damping. For example, let us consider to substitute (1) by the following procedure:

$$\mathbf{q}(k) = \mathcal{G}(K, \mathbf{p}(k), \mathbf{q}(k-1)) \quad (5a)$$

$$\mathbf{f}(k) = K(\mathbf{q}(k) - \mathbf{p}(k)) + B(\nabla\mathbf{q}(k) - \nabla\mathbf{p}(k))/T. \quad (5b)$$

Here,  $B > 0$  denotes the viscosity of the virtual coupling,  $T$  denotes the timestep size, and  $\nabla$  is the backward difference operator, which is defined as  $\nabla x(k) = x(k) - x(k-1)$ . This algorithm is used in, for example, [10]. It does not reflect the geometric relation between the proxy and the tool correctly to the reaction force  $\mathbf{f}(k)$ . For example, when a tool in contact with a virtual surface is pulled apart from the surface, the viscous term can produce an unnatural force sucking the tool toward the surface. This unnatural sucking (sticking) force has been recognized as one of the drawbacks of collision models based on the linear viscoelastic (Kelvin-Volgt) model [11].

Another approach for including damping is, as described in [8], for example, to replace (1) by

$$\mathbf{q}(k) = \mathcal{G}(K, \mathbf{p}(k), \mathbf{q}(k-1)) \quad (6a)$$

$$\mathbf{f}(k) = K(\mathbf{q}(k) - \mathbf{p}(k)) - B\nabla\mathbf{p}(k)/T. \quad (6b)$$

This algorithm represents the situation in which a grounded virtual damper is connected to the proxy. Thus, the user feels viscous resistance even when the tool is in the free space. It is not preferred because it produces unnecessary obstruction for the user's motion.

Admittance-type haptic rendering, on the other hand, requires a force measurement as the input. This scheme usually incorporates a virtual object representing the tool position in the virtual environment, and the motion of the object is updated depending on the force input. The virtual object in this case is similar to the proxy discussed in section 2.1, but it must have a nonzero mass  $M > 0$ . (Hereafter, we term it as a proxy also in admittance-type haptic rendering.) Let  $\mathbf{h}(k)$  denote the force measured by the force sensors, and  $\mathbf{f}(k)$  denote the force acting to the proxy from the virtual environment. Then, the proxy position  $\mathbf{q}(k)$  can be determined by the following equation of motion.

$$M\nabla^2\mathbf{q}(k)/T^2 = \mathbf{h}(k) + \mathbf{f}(k). \quad (7)$$

Here,  $\nabla^2$  satisfies  $\nabla^2 x(k) = \nabla x(k) - \nabla x(k-1)$ , and  $\nabla^2\mathbf{q}(k)/T^2$  represents the acceleration of the proxy. The

actual tool position does not appear in this equation because it is position-controlled to follow the proxy position  $\mathbf{q}(k)$ . The force  $\mathbf{f}(k)$  from the virtual environment is usually determined by the proxy motion in the virtual environment. In conclusion, it is not straightforward to use the geometric algorithm  $\mathcal{G}$  in admittance-type haptic rendering.

## 3. Proposals

### 3.1. Mechanistic Interpretation of Geometric Algorithms

In order to derive new haptic rendering techniques incorporating existing geometric algorithms  $\mathcal{G}$ , we discuss mechanistic interpretation of (1).

Equation (1b) yields  $\mathbf{p}(k) = \mathbf{q}(k) - \mathbf{f}(k)/K$ , and substituting it into (1a) yields the following:

$$\mathbf{q}(k) = \mathcal{G}(K, \mathbf{q}(k) - \mathbf{f}(k)/K, \mathbf{q}(k-1)). \quad (8)$$

This equation represents the algebraic constraint among  $K$ ,  $\mathbf{q}(k)$ ,  $\mathbf{q}(k-1)$ , and  $\mathbf{f}(k)$ . Here, we impose the following assumption:

**Assumption 1** *If  $\mathbf{r} = \mathcal{G}(K, \mathbf{r} - \mathbf{f}/K, \mathbf{q})$  is satisfied with vectors  $\mathbf{r} \in \mathbb{R}^n$ ,  $\mathbf{f} \in \mathbb{R}^n$ ,  $\mathbf{q} \in \mathbb{R}^n$ , and a positive real number  $K > 0$ , then  $\mathbf{r} = \mathcal{G}(\kappa, \mathbf{r} - \mathbf{f}/\kappa, \mathbf{q})$  is satisfied for all positive real numbers  $\kappa > 0$ .*

This means that whether (8) is satisfied or not is independent from  $K$ . When a positive real number  $T > 0$  is given, the assumption allows us to define a new function  $\mathcal{M} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  as follows:

$$\mathcal{M}(\mathbf{q}, \mathbf{v}) = \mathbf{f} \text{ s.t. } \mathbf{q} = \mathcal{G}(\kappa, \mathbf{q} - \mathbf{f}/\kappa, \mathbf{q} - T\mathbf{v}), \forall \kappa > 0. \quad (9)$$

This allows us to remove  $K$  from (8), showing that the following expression is algebraically equivalent to (8):

$$\mathbf{f}(k) = \mathcal{M}(\mathbf{q}(k), \nabla\mathbf{q}(k)/T). \quad (10)$$

Thus, **Assumption 1** allows the use of the following expression as an algebraically equivalent form of (1):

$$\mathbf{f}(k) = \mathcal{M}(\mathbf{q}(k), \nabla\mathbf{q}(k)/T) \quad (11a)$$

$$\mathbf{f}(k) = K(\mathbf{q}(k) - \mathbf{p}(k)). \quad (11b)$$

Although **Assumption 1** is not guaranteed to hold, its validity can be assumed as long as the function  $\mathcal{G}$  is mechanistically valid. The reason can be explained as follows. Because the proxy has no mass, the total force applied to the proxy is always zero. Equation (11b) implies that the force from the virtual coupling to the proxy is  $-\mathbf{f}(k)$ , and thus the sum of the forces from all other sources, i.e., the force from the virtual environment, is  $\mathbf{f}(k)$ . Therefore, the proxy

receives the force  $\mathbf{f}^{(k)}$  of (11a) from the virtual environment. Thus, (8) and (11a) can be interpreted as the mechanistic constraint resulting from the virtual environment, and therefore they must be independent from  $K$ , which is the parameter of the virtual coupling.

Equation (11) algebraically constrains two unknown values:  $\mathbf{f}^{(k)}$  and  $\mathbf{q}^{(k)}$ . Here, the function  $\mathcal{M}$  can be viewed as an impedance element that accepts the proxy motion (position  $\mathbf{q}^{(k)}$  and velocity  $\nabla\mathbf{q}^{(k)}/T$ ) as the input and provides the force  $\mathbf{f}^{(k)}$  from the virtual environment as the output.

As opposed to (9), when  $\mathcal{M}$  and  $T > 0$  are given, the correspondent  $\mathcal{G}$  can be obtained as follows:

$$\mathcal{G}(\kappa, \mathbf{p}, \mathbf{q}) = \mathbf{r} \text{ s.t. } \mathcal{M}(\mathbf{r}, (\mathbf{r} - \mathbf{q})/T) = \kappa(\mathbf{r} - \mathbf{p}). \quad (12)$$

### 3.2. Examples of Function $\mathcal{M}$

We describe the definitions of  $\mathcal{M}$  that correspond to the three examples of the function  $\mathcal{G}$  in section 2.1. In the first example of the unilateral constraint (Fig. 2 and (2)), the force  $\mathbf{f}$  from the virtual environment to the proxy and the proxy position  $\mathbf{q}$  satisfy the following condition:

$$(\mathbf{f} = \mathbf{o} \wedge \mathbf{n}^T \mathbf{q} > 0) \vee (\|\mathbf{f}\| = \mathbf{n}^T \mathbf{f} \wedge \mathbf{n}^T \mathbf{q} = 0). \quad (13)$$

Here,  $\mathbf{o}$  denotes the  $n$ -dimensional zero vector. Although (13) does not uniquely determine  $\mathbf{f}$  when  $\mathbf{n}^T \mathbf{q} = 0$ , the function  $\mathcal{M}$  can be described as the following set-valued form:

$$\mathcal{M}(\mathbf{q}, \dot{\mathbf{q}}) \begin{cases} = \mathbf{o} & \text{if } \mathbf{n}^T \mathbf{q} > 0 \\ \in \{\mathbf{f} \mathbf{n} \mid \mathbf{f} > 0\} & \text{if } \mathbf{n}^T \mathbf{q} = 0. \end{cases} \quad (14)$$

Using (14), we can replace (13) by  $\mathbf{f} = \mathcal{M}(\mathbf{q}, \dot{\mathbf{q}})$ .

In the case of the isotropic Coulomb friction (Fig. 3 and (3)), the relation between  $\mathbf{f}$  and  $\dot{\mathbf{q}}$  can be described as

$$(\mathbf{f} = -F\dot{\mathbf{q}}/\|\dot{\mathbf{q}}\| \wedge \dot{\mathbf{q}} \neq \mathbf{o}) \vee (\|\mathbf{f}\| \leq F \wedge \dot{\mathbf{q}} = \mathbf{o}). \quad (15)$$

This can be rewritten as  $\mathbf{f} = \mathcal{M}(\mathbf{q}, \dot{\mathbf{q}})$  under the following definition:

$$\mathcal{M}(\mathbf{q}, \dot{\mathbf{q}}) \begin{cases} = -F\dot{\mathbf{q}}/\|\dot{\mathbf{q}}\| & \text{if } \dot{\mathbf{q}} \neq \mathbf{o} \\ \in \{\mathbf{f} \in \mathbb{R}^n \mid \|\mathbf{f}\| \leq F\} & \text{if } \dot{\mathbf{q}} = \mathbf{o}. \end{cases} \quad (16)$$

Moreover, the function  $\mathcal{M}$  that represents the frictional flat surface (Fig. 4 and (4)) can be described as

$$\mathcal{M}(\mathbf{q}, \dot{\mathbf{q}}) \begin{cases} = \mathbf{o} & \text{if } \mathbf{n}^T \mathbf{q} > 0 \\ \in \left\{ \mathbf{f} \left( \mathbf{n} - \mu \frac{\mathbf{N}\mathbf{N}^T \dot{\mathbf{q}}}{\|\mathbf{N}^T \dot{\mathbf{q}}\|} \right) \mid \mathbf{f} > 0 \right\} & \text{if } \mathbf{n}^T \mathbf{q} = 0 \wedge \mathbf{N}^T \dot{\mathbf{q}} \neq \mathbf{o} \\ \in \left\{ \mathbf{f} (\mathbf{n} - \mu \mathbf{t}) \mid \mathbf{f} > 0, \mathbf{t}^T \mathbf{n} = 0, \|\mathbf{t}\| \leq 1 \right\} & \text{if } \mathbf{n}^T \mathbf{q} = 0 \wedge \mathbf{N}^T \dot{\mathbf{q}} = \mathbf{o} \end{cases} \quad (17)$$

where  $\mathbf{N}$  is a column full rank matrix that satisfies  $\mathbf{I} = \mathbf{N}\mathbf{N}^T + \mathbf{n}\mathbf{n}^T$  and  $\mathbf{N}^T \mathbf{n} = \mathbf{o}$  and  $\mathbf{I}$  is the  $n$ -dimensional identity matrix.

When the function  $\mathcal{G}$  is more complex, it is generally difficult to derive its correspondent function  $\mathcal{M}$ . However, the function  $\mathcal{M}$  is only used to provide a mechanistic interpretation to the geometric algorithm  $\mathcal{G}$ , and is not used for actual implementation.

The formulation  $\mathbf{f} = \mathcal{M}(\mathbf{q}, \dot{\mathbf{q}})$  can be discussed in relation to constraint-based simulation schemes [12, 13]. In such schemes, it is usual that the constraints in the whole system is described in well-established problem formulations such as Linear Complementarity Problems (LCPs), which can be numerically solved. Our formulation describes only the direct interaction between the proxy and the environment. Another difference is that the most of LCP-based simulation techniques includes constraints on the acceleration while our formulation,  $\mathbf{f} = \mathcal{M}(\mathbf{q}, \dot{\mathbf{q}})$ , does not.

### 3.3. Combining Function $\mathcal{G}$ and Damping in Impedance-Type Haptic Rendering

Equation (11) describes the virtual environment and the virtual coupling separately. Thus, adding a damping to the virtual coupling yields

$$\mathbf{f}^{(k)} = \mathcal{M}(\mathbf{q}^{(k)}, \nabla\mathbf{q}^{(k)}/T) \quad (18a)$$

$$\mathbf{f}^{(k)} = K(\mathbf{q}^{(k)} - \mathbf{p}^{(k)}) + B(\nabla\mathbf{q}^{(k)} - \nabla\mathbf{p}^{(k)})/T. \quad (18b)$$

This inclusion of damping is mechanically consistent.

Equation (18), which describes algebraic constraints between  $\mathbf{f}^{(k)}$  and  $\mathbf{q}^{(k)}$ , has a simple closed-form solution. It is easy to find (18b) is equivalent to the following:

$$\mathbf{f}^{(k)} = (K + B/T)(\mathbf{q}^{(k)} - \hat{\mathbf{p}}^{(k)}) \quad (19)$$

where

$$\hat{\mathbf{p}}^{(k)} = \mathbf{p}^{(k)} + \frac{B(\mathbf{q}^{(k-1)} - \mathbf{p}^{(k-1)})}{KT + B}. \quad (20)$$

Substituting (18a) by (19) yields

$$(K + B/T)(\mathbf{q}^{(k)} - \hat{\mathbf{p}}^{(k)}) = \mathcal{M}(\mathbf{q}^{(k)}, \nabla\mathbf{q}^{(k)}/T). \quad (21)$$

Because of (12), (21) is equivalent to the following:

$$\mathbf{q}^{(k)} = \mathcal{G}(K + B/T, \hat{\mathbf{p}}^{(k)}, \mathbf{q}^{(k-1)}). \quad (22)$$

Therefore, the analytical solution for (18) can be obtained by the following computational procedure:

$$\hat{\mathbf{p}}^{(k)} = \mathbf{p}^{(k)} + \frac{B(\mathbf{q}^{(k-1)} - \mathbf{p}^{(k-1)})}{KT + B} \quad (23a)$$

$$\mathbf{q}^{(k)} = \mathcal{G}(K + B/T, \hat{\mathbf{p}}^{(k)}, \mathbf{q}^{(k-1)}) \quad (23b)$$

$$\mathbf{f}^{(k)} = (K + B/T)(\mathbf{q}^{(k)} - \hat{\mathbf{p}}^{(k)}). \quad (23c)$$

The above computational procedure is obtained by replacing  $\mathbf{p}(k)$  and  $K$  by  $\hat{\mathbf{p}}(k)$  and  $K + B/T$ , respectively, in the basic equation (1) of the proxy-based haptic rendering. Developers do not need to care about the velocity feedback in using  $\mathcal{G}$ . Geometric algorithms in previous studies can be incorporated in (23) as long as it can be represented in the form of the function  $\mathcal{G}$ .

When the algorithm (23) is used with the function  $\mathcal{G}$  of (2), no sticking force can be produced when the tool is pulled apart from the surface. This is because the force  $\mathbf{f}$  becomes zero when  $\hat{\mathbf{p}}(k)$  comes above the surface even if the actual tool position  $\mathbf{p}(k)$  is still below the surface. It has been pointed out that the Kelvin-Volgt model has another drawback, which is the discontinuity in the force at the time of gaining contact [11]. This characteristic still remains in our method but practically it is not problematic in haptic rendering.

### 3.4. Using Function $\mathcal{G}$ in Admittance-Type Haptic Rendering

Based on the relation between  $\mathcal{M}$  and  $\mathcal{G}$  discussed in section 3.1, we can derive a new technique to incorporate the geometric algorithm  $\mathcal{G}$  in admittance-type haptic rendering. In admittance-type haptic rendering, the force  $\mathbf{h}$  applied from the user to the tool is measured by a force sensor, and the proxy position  $\mathbf{q}$  in the virtual environment is updated depending on  $\mathbf{h}$ . In this case, the proxy has a nonzero mass  $M$ , so the discrete-time representation of the equation of motion of the proxy can be described as follows:

$$M\nabla^2\mathbf{q}(k)/T^2 = \mathbf{h}(k) + \mathcal{M}(\mathbf{q}(k), \nabla\mathbf{q}(k)/T). \quad (24)$$

Equation (24) is equivalent to the following:

$$(M/T^2)(\mathbf{q}(k) - \hat{\mathbf{p}}(k)) = \mathcal{M}(\mathbf{q}(k), \nabla\mathbf{q}(k)/T) \quad (25a)$$

where

$$\hat{\mathbf{p}}(k) = 2\mathbf{q}(k-1) - \mathbf{q}(k-2) + T^2\mathbf{h}(k)/M. \quad (25b)$$

By using (12), we can rewrite (25) as follows:

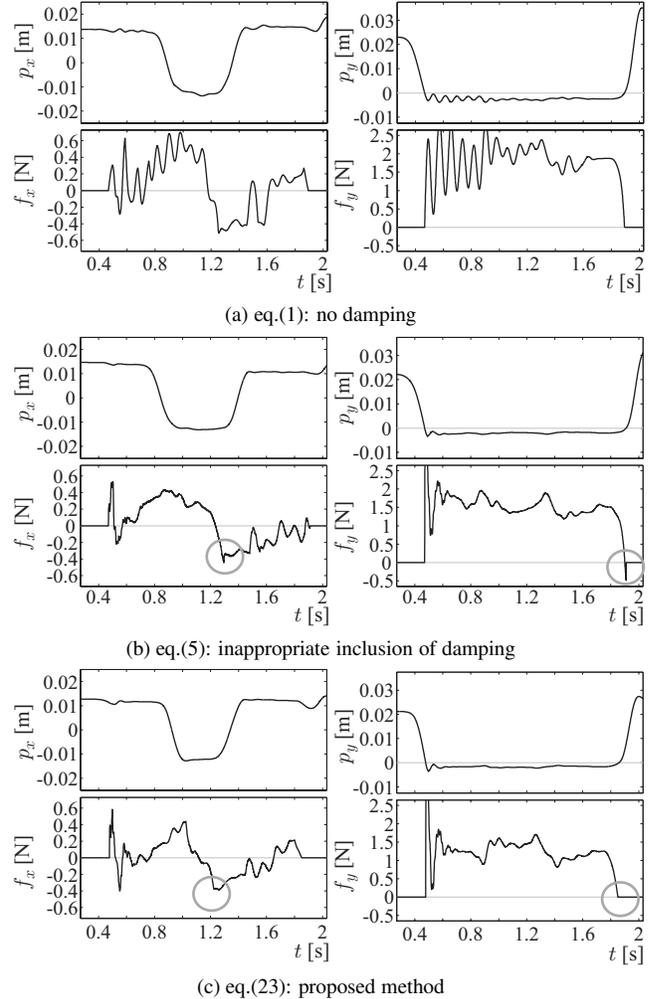
$$\hat{\mathbf{p}}(k) = 2\mathbf{q}(k-1) - \mathbf{q}(k-2) + T^2\mathbf{h}(k)/M \quad (26a)$$

$$\mathbf{q}(k) = \mathcal{G}(M/T^2, \hat{\mathbf{p}}(k), \mathbf{q}(k-1)). \quad (26b)$$

Thus, the geometric algorithm  $\mathcal{G}$  is incorporated in admittance-type haptic rendering by using  $\hat{\mathbf{p}}(k)$  of (26a) as the input to the function  $\mathcal{G}$ .

## 4. Implementation

The proposed method (23) of incorporating damping into impedance-type haptic rendering was demonstrated through



**Figure 5. Results of Experiment: impedance-type haptic rendering.**

a comparison to (5). (The other proposed technique (26) was not demonstrated because its validity is analytically shown and because the space of the paper is limited.) A PHANTOM<sup>®</sup> Desktop<sup>™</sup> device of SensAble Technologies was used, which is capable of 3-DOF actuation and 6-DOF position measurement. A virtual flat surface perpendicular to  $y$  axis was constructed in the virtual environment, and the friction coefficient of the surface was chosen as  $\mu = 0.3$ . The function  $\mathcal{G}$  in this case is described in (2), and it was implemented in three methods: (1), (5), and (23). Equation (1), which includes no damping, was implemented to show the damping effect of (23) to be as significant as that of (5). The parameters were set to be  $T = 0.001$  s,  $K = 750$  N/m, and  $B = 8$  Ns/m. The position measurements were smoothed with a first-order low-pass filter with the cut-off frequency of 30 Hz.

In the experiment, the tool was first pressed onto the vir-

tual surface, then was moved on the surface back and forth in  $x$  direction, and finally was moved apart from the surface in  $+y$  direction. Fig. 5 shows  $x$ - and  $y$ -components of the tool position  $\mathbf{p}$  and the actuator force  $\mathbf{f}$ . When the tool collided with the virtual surface (i.e., when  $p_y$  reached 0), the force in  $y$  direction,  $f_y$ , became oscillatory with (1), while not so with (5) and (23). This shows that the viscous damping works not only with (5) but also with (23). The negative  $x$  force observed at the collision is due to the static friction and fluctuations of the experimenter's hand intending to vertically press the tool onto the surface. The difference between (5) and (23) appears in the gray circles in Figs. 5(b) and (c). With (5), an impulse force in  $-y$  direction is produced when the tool was moving apart from the virtual surface. Moreover, another impulse force was produced in  $-x$  direction at the transition from static friction to kinetic friction. When (23) was used, such physically incorrect forces are not produced.

## 5. Conclusions

This paper has proposed two techniques for extending the scope of application of geometric (proxy-based) haptic rendering algorithms. One is the algorithm (23), which combines geometric algorithms and viscous damping in impedance-type haptic rendering. The other is the algorithm (26) to incorporate geometric algorithms in admittance-type haptic rendering. The results of implementation experiments have been presented.

The presented techniques are related to the discrete-time friction models for haptic rendering that the authors previously proposed [14]. Substituting the function  $\mathcal{G}$  of the definition (3) into (23) and (26) yield the impedance-type and admittance-type friction models described in [14].

While this paper has limited its scope to static virtual environments with stationary virtual objects, it will also be possible to treat a mobile virtual object by using a coordinate system attached to the object. A theoretical validation for this approach will be a part of our future research.

## References

- [1] M. Ueberle, N. Mock, and M. Buss, "VISHARD10, a novel hyper-redundant haptic interface," in *Proc. of 12th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004, pp. 58–65.
- [2] R. Q. Van Der Linde and P. Lammertse, "HapticMaster - a generic force controlled robot for human interaction," *Industrial Robot*, 30(6), pp. 515–524, 2003.
- [3] R. Taylor, P. Jensen, L. L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. de-Juan, and L. Kavoussi, "A steady-hand robotic system for microsurgical augmentation," *Int. J. of Robotics Research*, 18(12), pp. 1201–1210, 1999.
- [4] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. of 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 146–151.
- [5] D. C. Ruspini and O. Khatib, "Haptic display for human interaction with virtual dynamic environments," *J. of Robotic Systems*, 18(12), pp. 769–783, 2001.
- [6] V. Hayward and B. Armstrong, "A new computational model of friction applied to haptic rendering," in *Experimental Robotics VI*, 2000, pp. 404–412.
- [7] S. Hasegawa, N. Fujii, Y. Koike, and M. Sato, "Real-time rigid body simulation based on volumetric penalty method," in *Proc. of 11th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 326–332.
- [8] L. Kim, A. Kyrikou, G. Sukhatme, and M. Desbrun, "An implicit-based haptic rendering technique," in *Proc. of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002, pp. 2943–2948.
- [9] K. Lundin, A. Ynnerman, and B. Gudmundsson, "Proxy-based haptic feedback from volumetric density data," in *Proc. of EuroHaptics 2002*, 2002, pp. 104–109.
- [10] M. Ikits, J. D. Brederson, C. D. Hansen, and C. R. Johnson, "A constraint-based technique for haptic volume exploration," in *Proc. of 14th IEEE Visualization 2003 Conference*, 2003, pp. 263–269.
- [11] K. H. Hunt and F. R. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Trans. of ASME: J. of Applied Mechanics*, 7, pp. 440–445, 1975.
- [12] D. Baraff, "Fast contact force computation for non-penetrating rigid bodies," *Computer Graphics Proc., Annual Conference Series*, pp. 23–34, 1994.
- [13] M. Anitescu and F. A. Potra, "Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems," *Nonlinear Dynamics*, 14(3), pp. 231–247, 1997.
- [14] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Admittance and Impedance Representations of Friction Based on Implicit Euler Integration," *IEEE Trans. on Robotics*, 22(6), pp. 1176–1188, 2005.