Stable and Responsive Six-Degree-of-Freedom Haptic Manipulation Using Implicit Integration

Miguel A. Otaduy   Ming C. Lin

Department of Computer Science
University of North Carolina at Chapel Hill, USA
http://gamma.cs.unc.edu/interactive

Abstract

In this paper we present a novel approach for simulating the rigid body dynamics of a haptically manipulated object using implicit integration. Our formulation requires the linearization of contact and manipulation forces, and it provides higher stability and responsiveness than previous methods. The linearization of contact forces, coupled with fast, perceptually based collision detection algorithms, enables us to perform highly stable and responsive 6-degree-of-freedom haptic rendering of complex polygonal models.

1 Introduction

Humans use tactile and force cues to explore the environment around them and to identify and manipulate objects. The synthesis of force and torque feedback arising from object-object interaction, commonly referred to as six-degree-of-freedom (6-DoF) haptic rendering, can greatly benefit many applications involving dexterous manipulation and complex maneuvering of virtual objects. Examples of such applications include assembly and disassembly operations in rapid prototyping [17, 25], endoscopic surgical training [10, 14], and virtual exploration with limited visual feedback.

One of our research goals in computational haptics is to achieve stable and responsive 6-DoF haptic rendering of complex, rigid, polygonal models. The key to responsive and stable rendering is a very high force update rate [5, 7]. But, achieving a high force update rate becomes a difficult task with complex objects and complex contact scenarios, due to the inherent cost of collision detection.

We propose a haptic rendering algorithm based on implicit integration for penalty-based simulation that, coupled with fast, perceptually-based collision detection algorithms [22], enables stable and responsive 6-DoF haptic manipulation of complex polygonal models. Following previous approaches in haptic rendering, we decompose the rendering pipeline into two main modules: the simulation of the rigid body dynamics of the object grasped by the user, and the execution of collision detection and response. We also use two known techniques to interface between different modules. Virtual coupling [8, 2] handles the communication between the controller of the haptic device and the simulation of the grasped object, enabling bidirectional interaction. A linearized contact model serves as an intermediate representation [1] between the dynamic simulation and collision detection modules.

The main results of this paper include:

- Implicit integration for rigid body simulation with haptic interaction. This formulation involves the linearization of virtual coupling force and torque in the state space (i.e., positions and velocities) of a rigid body.
- A multirate rendering pipeline that decouples the simulation of the grasped object from collision detection and response, using a linearized penalty-based collision response model. Similarly, this formulation involves the linearization of penalty-based contact forces in the state space of a rigid body.
- Application of the 6-DoF haptic rendering algorithm to polygonal models with tens of thousands of triangles, achieving stable and responsive interaction under large contact stiffness and with small mass values.

The rest of this paper is organized as follows. Sec. 2 discusses related work, and Sec. 3 presents an overview of our rendering pipeline. Secs. 4, 5, and 6 describe the implicit integration of rigid body simulation, and the formulation of coupling and contact forces respectively. Sec. 7 presents the results. To conclude, Sec. 8 summarizes our work and discusses future research directions.

2 Related Work

In this section we discuss related work on the analysis of stability in haptic rendering and previous techniques for 6-DoF haptic rendering.

2.1 Stability in Haptic Rendering

Early stability analysis in haptic rendering focused on the problem of rendering stiff virtual walls. Several researchers [5, 7, 24] reached a common conclusion, that high force update rates are necessary in order to achieve stable rendering.

Later, Colgate et al. [8] proposed a multidimensional viscoelastic *virtual coupling* for stable interaction with nonlinear virtual environments. The stability of the system is guaranteed as long as each subsystem is itself passive. As noted by Colgate et al., one possible way to achieve a passive rigid body simulation is to combine implicit integration with penalty methods. Adams and Hannaford [2] extended the concept of virtual coupling by providing a unifying framework for impedance and admittance displays.

More recently, Miller et al. [18] have extended Colgate's passivity analysis techniques, relaxing the requirement of passive virtual environments but enforcing *cyclo-passivity* of the complete system. Hannaford et al. [12] have investigated the use of passivity observers and passivity controllers, instead of the traditional fixed-value virtual couplings.

Intermediate representations [1] have been very successful at improving the stability and responsiveness of haptic rendering systems. The general idea is to perform a full update of the virtual environment at a low frequency (limited by computational resources and the complexity of the system) and to use a simplified approximation for performing high-frequency updates of force feedback.

2.2 6-DoF Haptic Rendering

A number of previous techniques for 6-DoF haptic rendering follow the approach of *direct rendering*, for both parametric surfaces [19] and polygonal models [11, 16, 15]. In direct rendering, the position of the haptic device is applied directly to the grasped object, and collision response is computed as a function of object separation or penetration depth. The various techniques differ mostly in the algorithms used for accelerating collision detection. The main advantage of direct rendering is that there is no need to solve for the dynamics of the grasped object, and the dynamic behavior depends on the response of the user. However, penetration values may be quite large and visually perceptible, and system instability can arise if the frame rate of collision detection drops. Recently, Johnson and Willemsen [15] have incorporated an approximate but fast, incremental contact-point-tracking algorithm that is combined with slower exact collision updates. This algorithm handles models with thousands of triangles at interactive rates, but the forces may suffer discontinuities if the exact update is too slow.

Other previous techniques for 6-DoF haptic rendering combine virtual coupling [8] with rigid body simulation of the grasped object. The advantages of virtual coupling techniques are reduced interpenetration, higher stability, and higher control of the displayed stiffness. The main disadvantages are that the coupling may introduce noticeable filtering, both tactile and visual, and that the dynamics of the grasped object must be simulated. Some researchers have proposed the combination of virtual coupling with constraint-based simulation [4, 23] or impulse-based simulation [6, 9]. However, these approaches have been tested only on rather simple benchmarks. Others have combined virtual coupling with penalty-based collision response [17, 25]. McNeely et al. [17] proposed a combination of point-sampling and voxelization for solving the problem of collision detection. They alleviated some of the penetrability and stability problems of their approach by applying pre-contact braking forces and by averaging contact forces. Later, Wan and McNeely [25] used the same technique for collision detection, but computed the position of the grasped object following a quasi-static approximation. They linearized the coupling and contact force and torque and solved for the position and orientation of equilibrium.

The application of 6-DoF haptic rendering algorithms to complex models and complex contact scenarios becomes a challenging issue, due to the inherent cost of collision detection that induces slow force updates. Otaduy and Lin [22] have presented a sensation-preserving simplification technique for 6-DoF haptic rendering of complex polygonal models by selecting contact resolutions adaptively. Otaduy et al. [21] have also proposed a rendering algorithm for the interaction of textured surfaces. Their work is focused on the acceleration of collision detection and response, but they rely on previous techniques for displaying force-and-torque feedback.

3 Overview

As stated in the introduction, our approach to 6-DoF haptic rendering combines virtual coupling with penalty-based rigid-body dynamic simulation of the object grasped by the user. In order to achieve stable and responsive interaction, it is especially important to maintain a high force update rate. In this regard, penalty-based methods offer important advantages over other techniques for simulating rigid body dynamics. However, penalty-based methods cannot enforce non-penetration, but with high contact stiffness object interpenetration can be small or minimized by using small collision tolerances. Implicit integration of rigid body dynamics provides passivity [8] and higher stability under high contact stiffness values [3].

By using a linearized contact model, we decompose the haptic rendering pipeline into two threads: a *haptic thread* that performs the rigid-body dynamic simulation of the

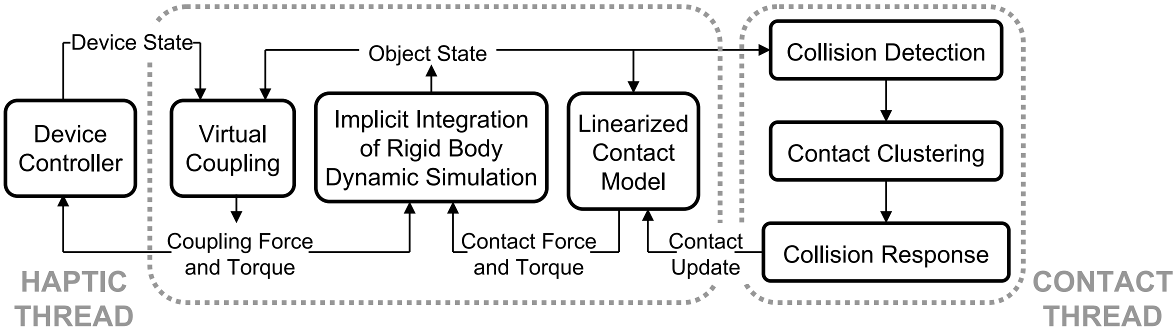

Figure 1: **Rendering Pipeline.** *A haptic thread runs at force update rates simulating the dynamics of the grasped object and computing force feedback, while a contact thread runs asynchronously and updates contact forces.*

grasped object, and a *contact thread* that executes collision detection and response. In this way, collision detection is less a bottleneck for the simulation and the synthesis of feedback force and torque. The different threads and modules of the rendering pipeline are highlighted in Fig. 1.

Next we describe the threads of the rendering pipeline in more detail, and we introduce the notation used throughout the paper.

3.1 Multirate Architecture

The **haptic thread** runs at a high frequency (1kHz in the experiments described in Sec. 7), computing rigid body simulation and force feedback. Each frame, the haptic thread executes the following sequence of operations:

1. Read state of the haptic device at time t_i.
2. Linearize the coupling force and torque at time t_{i-1}.
3. Linearize the contact force and torque at time t_{i-1}.
4. Solve the state of the grasped object at time t_i, using implicit integration.
5. Compute the coupling force and torque at time t_i.
6. Send the coupling force and torque to the device controller.

The **contact thread** runs asynchronously, at the highest frequency possible, given the complexity of the contact scenario. We have followed the sensation-preserving simplification approach proposed by Otaduy and Lin [22] for executing collision detection between complex polygonal models. Specifically, the contact thread performs the following sequence of operations every loop:

1. Fetch the state of the grasped object.
2. Perform collision detection based on sensation-preserving simplification.
3. Cluster contacts and compute cluster representatives.
4. For each cluster representative, compute the contact force and torque and their Jacobians.

3.2 Notation

We use lower-case bold-face letters to represent vectors and quaternions, and upper-case letters to represent matrices. In matrix operations, vectors are in column form, and quaternions are treated as 4×1 vectors, unless we explicitly indicate that they are involved in quaternion products. Unless otherwise specified, all magnitudes are expressed in global coordinates of the virtual world. Given a vector $\mathbf{u} = (u_x, u_y, u_z)^T$, $\mathbf{u}^*$ denotes the skew-symmetric matrix used for representing a cross product as a matrix-vector product:

$$\mathbf{u}^* = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \tag{1}$$

4 Rigid Body Dynamics

In this section, we formulate the implicit integration for penalty-based dynamic simulation of the grasped object.

4.1 Equations of Rigid Body Motion

We formulate the state of a rigid body in terms of the position of its center of mass, $\mathbf{x}$, a quaternion describing its orientation, $\mathbf{q}$, its linear momentum, $\mathbf{P}$, and its angular momentum, $\mathbf{L}$. With this selection of state variables, the Newton-Euler equations that describe rigid body dynamics can be written as a function of external forces $\mathbf{F}$ and torques $\mathbf{T}$ by the following ODEs:

$$\dot{\mathbf{y}}(t) = \begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{L}} \end{pmatrix} = \begin{pmatrix} \frac{1}{m}\mathbf{P} \\ \frac{1}{2}\omega_q \mathbf{q} \\ \mathbf{F} \\ \mathbf{T} \end{pmatrix} \tag{2}$$

where m is the mass of the body. The term ω_q indicates a quaternion with scalar part 0 and vector part the angular velocity ω. Given the rotation matrix R and the mass matrix M of the body, its angular velocity ω can be expressed in terms of state variables as:

$$\omega = RM^{-1}R^T\mathbf{L} \tag{3}$$

In many of the practical applications of 6-DoF haptic rendering (e.g., assembly and disassembly tasks or surgical operations on hard tissue), the environment can be considered to be static. Following this observation, and as many others have done in the past [19, 16, 15, 17, 25], we assume that the only moving object in the simulation is the grasped object. With this assumption, the state vector $\mathbf{y}$ has 13 variables. The external forces (and similarly for the torques) comprise the weight of the object, penalty-based contact forces $\mathbf{F}_p$, and the virtual coupling force $\mathbf{F}_c$. Other terms, such as friction, could also be added. The simplest way to incorporate friction into the formulation of external forces would be by using a local friction model [13].

4.2 Implicit Integration

The system of ODEs describing rigid body motion can be represented in a vector form as:

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t) \tag{4}$$

Implicit discretization of the ODEs using the Backward Euler formula yields the following equation for the update of the state vector:

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t \dot{\mathbf{y}}_n \tag{5}$$

Substituting Eq. 2 in Eq. 5 leads to a nonlinear equation in the state variables $\mathbf{x}$, $\mathbf{q}$, $\mathbf{P}$ and $\mathbf{L}$. A nonlinear solver, such as Newton's method, can be used for finding the exact solution to this system. However, we have decided to trade accuracy for speed, and linearly approximate Eq. 5 using the Taylor expansion of $\mathbf{f}$. This approximation leads to a semi-implicit Backward Euler discretization, in which $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}$ is the Jacobian of the equations of rigid body motion. Rearranging terms, the linear system of equations can be expressed in the form:

$$\left(I - \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right) (\mathbf{y}_n - \mathbf{y}_{n-1}) = \Delta t \mathbf{f}_{n-1} \tag{6}$$

Under the assumption that the grasped object is the only moving object, $\left(I - \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right)$ is a 13×13 dense and non-symmetric matrix. The linear system can be solved by Gaussian elimination. The remaining of this section focuses on the formulation of the Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}$.

4.3 Jacobian of Rigid Body Motion

From Eq. 2, the Jacobian can be expressed as:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{y}} = \begin{pmatrix} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} & \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{P}} & \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{L}} \\ \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{x}} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{P}} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{L}} \\ \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{x}} & \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{P}} & \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{L}} \\ \frac{\partial \dot{\mathbf{L}}}{\partial \mathbf{x}} & \frac{\partial \dot{\mathbf{L}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{L}}}{\partial \mathbf{P}} & \frac{\partial \dot{\mathbf{L}}}{\partial \mathbf{L}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{m} I & 0 \\ 0 & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} & 0 & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{L}} \\ \frac{\partial \mathbf{F}}{\partial \mathbf{x}} & \frac{\partial \mathbf{F}}{\partial \mathbf{q}} & \frac{\partial \mathbf{F}}{\partial \mathbf{P}} & \frac{\partial \mathbf{F}}{\partial \mathbf{L}} \\ \frac{\partial \mathbf{T}}{\partial \mathbf{x}} & \frac{\partial \mathbf{T}}{\partial \mathbf{q}} & \frac{\partial \mathbf{T}}{\partial \mathbf{P}} & \frac{\partial \mathbf{T}}{\partial \mathbf{L}} \end{pmatrix} \tag{7}$$

The evaluation of the Jacobian requires the computation of the Jacobians of external forces (and torques). Sections 5 and 6 deal, respectively, with coupling forces and contact forces.

The expression of the derivative of orientation, $\dot{\mathbf{q}}$, is highly nonlinear and leads to two non-zero blocks in the Jacobian, as shown in Eq. 7. The expression of $\dot{\mathbf{q}}$ can be rewritten as a matrix-vector multiplication, with a matrix Q formed with the coefficients of $\mathbf{q}$:

$$\dot{\mathbf{q}} = \frac{1}{2} \omega_q \mathbf{q} = Q \omega \tag{8}$$

Combining Eqs. 3 and 8, we obtain the following Jacobians:

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{L}} = Q R M^{-1} R^T \tag{9}$$

$$\frac{\partial \dot{\mathbf{q}}}{\partial q_i} = \frac{\partial Q}{\partial q_i} \omega + Q \frac{\partial \omega}{\partial q_i} \tag{10}$$

$$\frac{\partial \omega}{\partial q_i} = \left(\frac{\partial R}{\partial q_i} M^{-1} R^T + R M^{-1} \frac{\partial R}{\partial q_i}^T \right) \mathbf{L} \tag{11}$$

Note that $\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}$ is expressed separately for each of the components q_i of $\mathbf{q}$. For more details on the formulation of the Jacobian of $\dot{\mathbf{q}}$ and the derivatives $\frac{\partial Q}{\partial q_i}$ and $\frac{\partial R}{\partial q_i}$, please refer to [20].

5 Virtual Coupling

In this section, we describe the equations for coupling force and torque that enable bidirectional interaction with a grasped object. We also list the Jacobians of coupling force and torque, which are used in the implicit integration of rigid body dynamic simulations, and we discuss issues associated with the synthesis of force feedback from a virtual coupling.

5.1 Coupling Force and Torque

When an object is grasped, the state of the haptic device in the virtual world is recorded as a coupling frame (coupling position $\mathbf{c}$ and coupling orientation $\mathbf{q}_c$) in the local coordinates of the object. During manipulation, the coupling force $\mathbf{F}_c$ is set as a viscoelastic link between the current position of the haptic device $\mathbf{x}_h$ and the position of the coupling point. The coupling torque $\mathbf{T}_c$ is composed of the torque induced by the coupling force, and a viscoelastic rotational link between the current orientation of the haptic device $\mathbf{q}_h$ and the current orientation of the coupling frame. The rotational link can be expressed in terms of its equivalent axis of rotation, $\mathbf{u_c}$. The magnitude of $\mathbf{u_c}$ represents the coupling angle. The coupling force and torque equations are:

$$\begin{aligned} \mathbf{F}_c &= k_c(\mathbf{x}_h - \mathbf{x} - R\mathbf{c}) + b_c(\mathbf{v}_h - \mathbf{v} - \omega \times \mathbf{c}) \\ \mathbf{T}_c &= (R\mathbf{c}) \times \mathbf{F}_c + k_\theta \mathbf{u_c} + b_\theta(\omega_h - \omega) \end{aligned} \tag{12}$$

where k_c and b_c represent linear stiffness and damping respectively; k_θ and b_θ represent angular stiffness and damping respectively; and $\mathbf{x}_h$, $\mathbf{v}_h$, and ω_h represent the position, linear velocity, and angular velocity of the haptic device.

5.2 Jacobian of Virtual Coupling

Here we list the Jacobians of coupling force and torque w.r.t. the different state variables. Note that the Jacobians w.r.t. the quaternion are expressed columnwise (i.e., separately for each component q_i of the quaternion). For more details on the formulation of the Jacobians, please refer to [20].

$$\frac{\partial \mathbf{F}_c}{\partial \mathbf{x}} = -k_c I \tag{13}$$

$$\frac{\partial \mathbf{T}_c}{\partial \mathbf{x}} = -k_c (R\mathbf{c})^* \tag{14}$$

$$\frac{\partial \mathbf{F}_c}{\partial q_i} = -k_c \frac{\partial R}{\partial q_i}\mathbf{c} + b_c \mathbf{c}^* \frac{\partial \omega}{\partial q_i} \tag{15}$$

$$\frac{\partial \mathbf{T}_c}{\partial q_i} = (R\mathbf{c})^* \frac{\partial \mathbf{F}_c}{\partial q_i} - \mathbf{F}_c{}^* \frac{\partial R}{\partial q_i}\mathbf{c} + k_\theta \frac{\partial \mathbf{u_c}}{\partial q_i} - b_\theta \frac{\partial \omega}{\partial q_i} \tag{16}$$

$$\frac{\partial \mathbf{F}_c}{\partial \mathbf{P}} = -\frac{b_c}{m} I \tag{17}$$

$$\frac{\partial \mathbf{T}_c}{\partial \mathbf{P}} = -\frac{b_c}{m} (R\mathbf{c})^* \tag{18}$$

$$\frac{\partial \mathbf{F}_c}{\partial \mathbf{L}} = b_c \mathbf{c}^* R M^{-1} R^T \tag{19}$$

$$\frac{\partial \mathbf{T}_c}{\partial \mathbf{L}} = \left(b_c (R\mathbf{c})^* \mathbf{c}^* - b_\theta I\right) R M^{-1} R^T \tag{20}$$

5.3 Synthesis of Force Feedback

After solving the object state at each frame, we compute coupling force and torque based on Eq. 12, using the newly computed object state. The resulting force and torque values are sent to the device controller as feedback commands.

5.3.1 Nonlinear Coupling

Haptic devices present physical limitations that should also be accounted for in the design of virtual coupling. Force (and torque) saturation is one example. When the user pushes against a virtual surface and the device reaches its maximum force value, the user feels no difference as a result of pushing further. The coupling force in the simulation, however, keeps growing; and so does object interpenetration. To avoid this, we suggest modeling the coupling stiffness as a nonlinear function, in a way similar to Wan and McNeely [25]. We propose a spline force function: (1) for small deviations, under the saturation value, a linear force equation; (2) a cubic interpolating force equation; and (3) for large deviations, a constant saturated force. The coupling Jacobian must be revised, to account for the nonlinearity of the stiffness. Please refer to [20] for more details.

6 Collision Detection and Response

We begin this section by describing the contact information returned by the collision detection module, and then we describe the force and torque equations for collision response, as well as their Jacobians. We conclude the section with the formulation of the linearized contact model.

6.1 Collision Detection

We perform collision detection using the sensation-preserving simplification algorithm proposed by Otaduy and Lin [22]. A contact query returns a set of contacts that sample the regions of the objects that are closer than a distance tolerance d. Each contact is described by a point $\mathbf{p}$ on the surface of the grasped object, a point $\mathbf{p}_0$ on the surface of the object in the scene, the contact normal $\mathbf{n}$ pointing outward from the grasped object, and the penetration depth δ (which is positive if $\mathbf{p}$ lies inside the scene object, and negative if $\mathbf{p}$ lies outside, but closer than d).

A contact query may return multiple contacts to describe each contact region. As pointed out by others earlier [17, 16], discontinuities in the number of contacts affect the stability of penalty-based simulations, because the total stiffness depends on the number of contacts. We cluster contacts based on the K-means clustering technique, and we compute a representative contact per cluster. Please refer to [20] for more details.

6.2 Penalty-Based Collision Response

After contact clustering, the contact normal $\mathbf{n}$ is a representative value that does not capture exact information about surface features, therefore we have opted to model each contact as a planar constraint. The constraint is represented by the plane with normal $\mathbf{n}$ and passing through $\mathbf{p}_0$. Note that it is also convenient to represent $\mathbf{p}$ based on its coordinates in the local frame of the grasped object, $\mathbf{r}$. We compute viscoelastic penalty-based force $\mathbf{F}_p$ and torque $\mathbf{T}_p$ as:

$$\begin{aligned} \mathbf{F}_p &= -kN(\mathbf{x} + R\mathbf{r} - \mathbf{p}_0) - kd\mathbf{n} - bN(\mathbf{v} + \omega \times \mathbf{r}) \\ \mathbf{T}_p &= (R\mathbf{r}) \times \mathbf{F}_p \end{aligned} \tag{21}$$

N is a matrix that projects a vector onto the normal of the constraint plane, and it is computed as $\mathbf{n}\,\mathbf{n}^T$.

6.3 Jacobian of Contact Force and Torque

Here we list the Jacobians of penalty-based force and torque w.r.t. the different state variables. Note that the

Jacobians w.r.t. the quaternion are expressed columnwise, and the contact normal is considered to be constant during one frame of the simulation. For more details in the formulation of the Jacobians, please refer to [20].

$$\frac{\partial \mathbf{F}_p}{\partial \mathbf{x}} = -kN \tag{22}$$

$$\frac{\partial \mathbf{T}_p}{\partial \mathbf{x}} = (R\mathbf{r})^* \frac{\partial \mathbf{F}_p}{\partial \mathbf{x}} \tag{23}$$

$$\frac{\partial \mathbf{F}_p}{\partial q_i} = -kN\frac{\partial R}{\partial q_i}\mathbf{r} - bN\omega^*\frac{\partial R}{\partial q_i}\mathbf{r} + bN(R\mathbf{r})^*\frac{\partial \omega}{\partial q_i} \tag{24}$$

$$\frac{\partial \mathbf{T}_p}{\partial q_i} = (R\mathbf{r})^* \frac{\partial \mathbf{F}_p}{\partial q_i} - \mathbf{F}_p{}^* \frac{\partial R}{\partial q_i}\mathbf{r} \tag{25}$$

$$\frac{\partial \mathbf{F}_p}{\partial \mathbf{P}} = -\frac{b}{m}N \tag{26}$$

$$\frac{\partial \mathbf{T}_p}{\partial \mathbf{P}} = (R\mathbf{r})^* \frac{\partial \mathbf{F}_p}{\partial \mathbf{P}} \tag{27}$$

$$\frac{\partial \mathbf{F}_p}{\partial \mathbf{L}} = bN(R\mathbf{r})^* RM^{-1}R^T \tag{28}$$

$$\frac{\partial \mathbf{T}_p}{\partial \mathbf{L}} = (R\mathbf{r})^* \frac{\partial \mathbf{F}_p}{\partial \mathbf{L}} \tag{29}$$

6.4 Linearized Contact Model

In complex contact configurations, collision detection may easily run at rates notably slower than the update of rigid body dynamics, even with sensation-preserving simplification [22]. In such cases, linear approximations of the contact forces increase the accuracy of the derivatives of state variables, and thereby the stability of implicit integration. Assuming that the last update of contact force (and similarly for the torque) took place at time t, the contact force $\mathbf{F}_p$ at time $t+\Delta t$ can be linearly approximated using its Taylor expansion as:

$$\mathbf{F}_p(t+\Delta t) = \mathbf{F}_p(t) + \frac{\partial \mathbf{F}_p}{\partial \mathbf{y}}(t)\,(\mathbf{y}(t+\Delta t) - \mathbf{y}(t)) \tag{30}$$

Note that penalty-based contact forces depend solely on the state of the grasped object, therefore $\frac{\partial \mathbf{F}_p}{\partial t} = 0$ and $\frac{\partial \mathbf{T}_p}{\partial t} = 0$. The Jacobians of contact forces and torques w.r.t. state variables $\mathbf{y}$ must also be computed for the semi-implicit formulation of Backward Euler. Therefore, the computation of the linearized contact model has little additional cost.

7 Experiments and Results

In this section, we describe implementation details and experiments designed to evaluate the responsiveness and stability of the 6-DoF haptic rendering algorithm.

7.1 Implementation Details

The experiments have been performed using a dual Pentium-4 2.4GHz processor PC with 2.0 GB of memory and an NVidia GeForce FX5950 graphics card, and Windows2000 OS. We have used a 6-DoF *Phantom*^{TM} impedance-type haptic device, but our formulation is also applicable to admittance-type haptic devices, following Adams and Hannaford's framework [2]. The haptic thread is executed at a constant frequency of 1kHz, and it employs utilities of GHOST-SDK, the software API of the *Phantom*^{TM} haptic device, to communicate with the device controller. The contact thread is executed asynchronously and is assigned a lower scheduling priority.

7.2 Analysis of Free-Space Motion

We have designed an experiment to evaluate the performance of implicit integration for rigid body dynamics simulation during free-space motion with virtual coupling. In the experiment, the haptic device commands the motion of a 20cm-long spoon (see Fig. 3). The spoon is moved freely, without touching other objects. A thin object, such as a spoon, is particularly challenging for numerical integration due to its low inertia around its longitudinal axis.

Fig. 2 reflects the coupling deviation, $\|\mathbf{x}_h - \mathbf{x}_c\|$, and the absolute value of coupling force, $\|\mathbf{F}_c\|$, during 2.5 seconds of simulation. We have collected the values of coupling deviation and force using different numerical integration methods (i.e., Forward Euler, Runge-Kutta IV, and Backward Euler) and the same pre-recorded trajectory of the haptic device. Using the Backward Euler implicit integration method, with coupling stiffness $k_c = 200$N/m and $k_\theta = 0.6$Nm/rad, the simulation is stable with a mass as small as 1g. However, using explicit integration methods, such as Runge-Kutta IV or Forward Euler, the simulation is stable only with masses larger than 70g and 100g respectively.

The left graph of Fig. 2 shows the coupling deviation, which reaches 17mm with a mass of 100g, but it never exceeds 2mm with a mass of 10g. The logarithmic plot in Fig. 2 indicates that the coupling deviation is roughly linear w.r.t. the mass of the spoon. The right graph of Fig. 2 shows the coupling force, which reaches 3.5N with a mass of 100g, but it never exceeds 0.5N with a mass of 10g. The results of the experiment indicate that, for the same coupling stiffness, the coupling deviation is larger when the mass of the grasped object is larger. Similarly, the coupling force is also larger when the mass is larger. From these two observations, and considering that stable mass values are substantially larger with explicit integration, we conclude that implicit integration enables more responsive free-space interaction with virtual coupling than explicit integration.

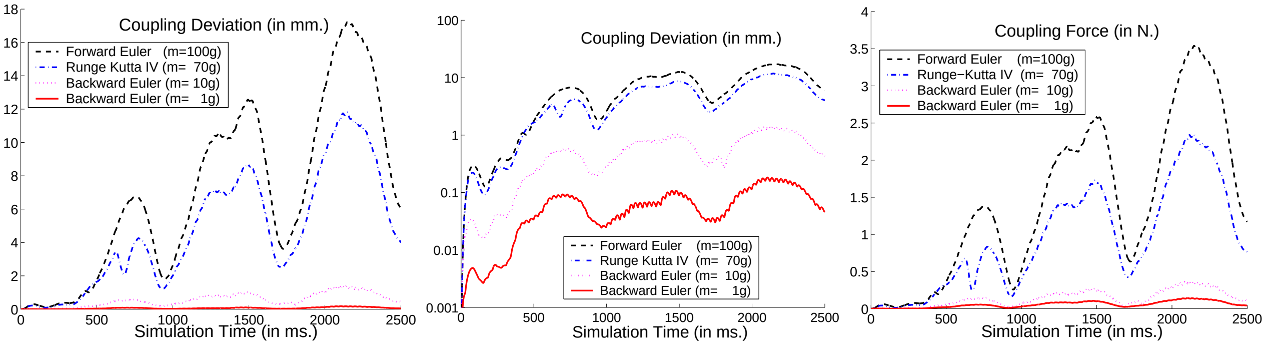

Figure 2: **Coupling Deviation and Force During Free-Space Motion.** *Comparisons using different numerical integration methods, and varying the mass of the grasped object. Left: deviation between the position of the haptic device and the position of the coupling point in the spoon; Center: log plot of the coupling deviation; Right: coupling force.*

7.3 Analysis of Contact State

A scenario with relatively simple models (i.e., the cup and the spoon depicted in Fig. 3) has been used to evaluate the stability and responsiveness of the haptic rendering approach, by integrating implicit integration with penalty-based methods and virtual coupling. We recorded a trajectory of the haptic device while manipulating a virtual spoon ($1,344$ triangles and 20cm-long) in contact with a virtual cup ($4,000$ triangles and 8cm-radius). Then, we played this trajectory using different haptic rendering settings. We analyzed the stability and responsiveness of the system with different contact stiffness values and with different integration methods. Fig. 4 shows graphs of maximum local penetration depth (top left), coupling deviation (top right), contact force (bottom left), and feedback or coupling force (bottom right) during 650 milliseconds of the simulation and the following settings: (1) Runge-Kutta IV, $m = 100$g, and $k = 2$kN/m; (2) Backward Euler, $m = 10$g, and $k = 2$kN/m; and (3) Backward Euler, $m = 100$g, and $k = 10$kN/m. The coupling stiffness is 200N/m in all three cases.

As can be inferred from the graph of penetration depth in Fig. 4, the spoon moved in free-space for a period of more than 100 milliseconds, and then started penetrating the surface of the cup. The spoon remained in contact with the cup (penetrating slightly) during the rest of the simulation.

Numerical integration of the simulation of the spoon with the Runge-Kutta IV method is stable for values of the mass larger than 70g, as concluded from the analysis of free-space motion. This requirement affects the performance during contact state as well. As reflected in the bottom right graph of Fig. 4, with a mass of 100g, the magnitude of feedback force during free-space motion and contact situations is very similar. This similarity degrades the kinesthetic perception of contact. Implicit integration is stable for small values of the mass. This result shows a clear distinction in the magnitude of feedback force between free-space motion and contact state.

High contact stiffness minimizes the amount of interpenetration between the spoon and the cup. As shown in the top left graph of Fig. 4, the maximum local penetration during the interval of study was smaller than 0.6mm with a contact stiffness of 2kN/m, and smaller than 0.2mm with a contact stiffness of 10kN/m. As a conclusion, penalty-based collision response with high contact stiffness enables small visual interpenetrations, which can enhance the perception of hard contact.

The numerical integration of the dynamic simulation of the spoon is susceptible to instability problems with high contact stiffness. Contact clustering alleviates the discontinuities of contact-point positions, but (smaller) discontinuities are still present. And, they may induce large oscillations of the contact force and the penetration depth, as shown in the left graphs of Fig. 4. Note the existence of oscillations with Runge-Kutta IV and $k = 2$kN/m, and with Backward Euler and $k = 10$kN/m. Out of the interval of study, the oscillations with these settings became more serious, and were also transmitted to the coupling force. However, with Backward Euler and $k = 2$kN/m, the numerical integration of the dynamic simulation of the spoon remained stable. Implicit integration methods enable stable penalty-based rigid body simulation with (relatively) high contact stiffness and small mass values.

7.4 Analysis with Complex Models

A scenario with two complex virtual jaws (See Fig. 5) has been used to test the effectiveness of the linearized contact model and the stability and responsiveness of our haptic rendering algorithm on complex polygonal models. The model of the lower jaw is composed of $40,180$ triangles, while the upper jaw consists of $47,339$ triangles.

We recorded a trajectory of the upper jaw while rendering the interaction with the lower jaw and using sensation-preserving simplification [22] with an error threshold of 2.5% of the radius of the jaws. Then, we played this same trajectory with smaller error thresholds of 1% and 0.4%, thereby increasing the cost of collision detection and decreasing the update rate of the contact thread. We ran the

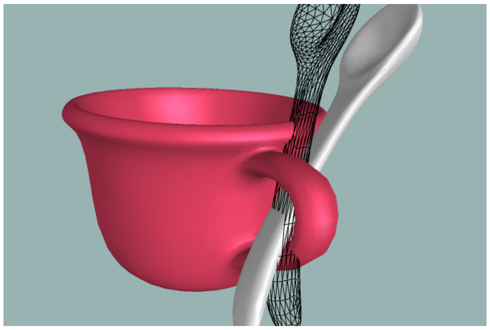

Figure 3: **Manipulation of a Spoon in Contact with a Cup Using Virtual Coupling.** *As the spoon is constrained inside the handle of the cup, the contact force and torque are perceived through a virtual coupling. A wireframe image of the spoon represents the actual configuration of the haptic device.*

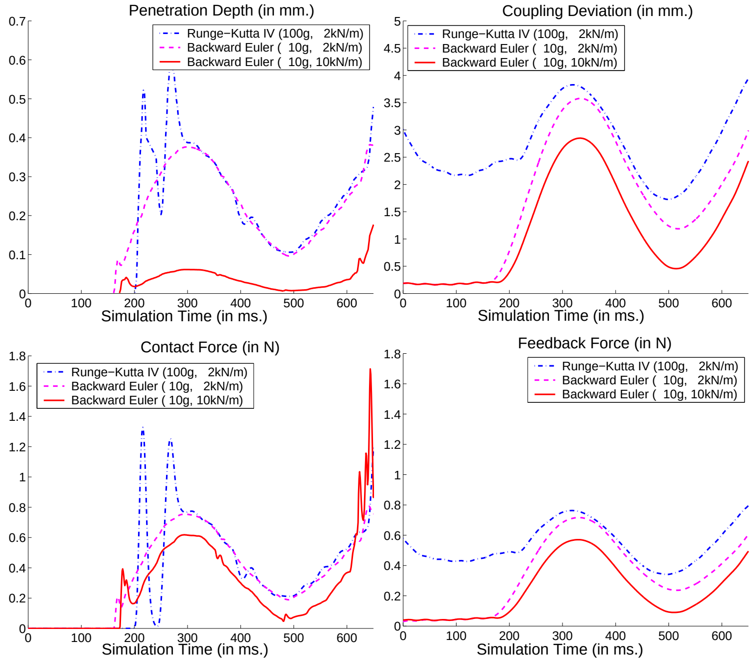

Figure 4: **Analysis of Forces and Positions During Contact.** *Comparison of maximum local penetration depth (top left), coupling deviation (top right), contact force (bottom left), and feedback or coupling force (bottom right) using different numerical integration methods and contact stiffness values.*

experiments with and without the use of the linearized contact model. In the experiment without linearized contact model and with an error threshold of 0.4%, the simulation soon became unstable to the point that the state of the upper jaw diverged to infinity. For clarity of the graphs, we have not included the data of this experiment.

Fig. 6 shows graphs of maximum local penetration depth (top left), frame rate of the contact thread (bottom left), coupling deviation (center), and feedback or coupling force (right) during 900 milliseconds of simulation, using different error tolerances for sensation-preserving simplification, with and without (w/o) linearized contact model. The models of both jaws can be bounded by spheres of 6cm-radius. We scaled the workspace of the haptic device by a factor of 0.4, therefore the forces plotted in the graphs are scaled by a factor of 2.5 before being fed back to the user. All the experiments were executed using Backward Euler semi-implicit integration as described in Sec. 4.2, a mass $m = 10$g for the upper jaw, coupling stiffness $k_c = 500$N/m, and contact stiffness $k = 5$kN/m.

The plots demonstrate that, with the linearized contact model and an error threshold of 2.5% the behavior of the system became very stable and responsive. For example, the maximum local penetration depth never exceeded 0.1mm, thanks to high stability with a contact stiffness as high as 5kN/m. With the linearized contact model but reducing the error threshold, the behavior degraded slightly, but remained considerably stable. With an error threshold of 0.4% the update rate of the contact thread went down to 100Hz at times. Even in such a challenging situation, the computation of approximate contact forces with the linearized contact model maintained high stability.

On the other hand, without the linearized contact model, the performance degraded rapidly. Even with an error threshold of 2.5%, which kept the update rate of the contact thread over 500Hz., the feedback force became clearly unstable. The comparison of simulation data with and without the linearized contact model clearly indicates the influence of the linearized contact model on the stability of the system when the update rate of the contact thread decays. This observation demonstrates that the linearized contact model is a key factor for successful 6-DoF haptic rendering of complex models.

8 Conclusion

We have presented a novel approach for 6-DoF haptic rendering, by simulating the rigid body dynamics of the grasped object using implicit integration. Implicit integration involves the linearization of virtual coupling and penalty-based force and torque in the state space of the rigid body. We have combined our approach with a fast, perceptually-based collision detection algorithm [22], producing stable and responsive haptic manipulation of objects with tens of thousands of triangles. Next we compare

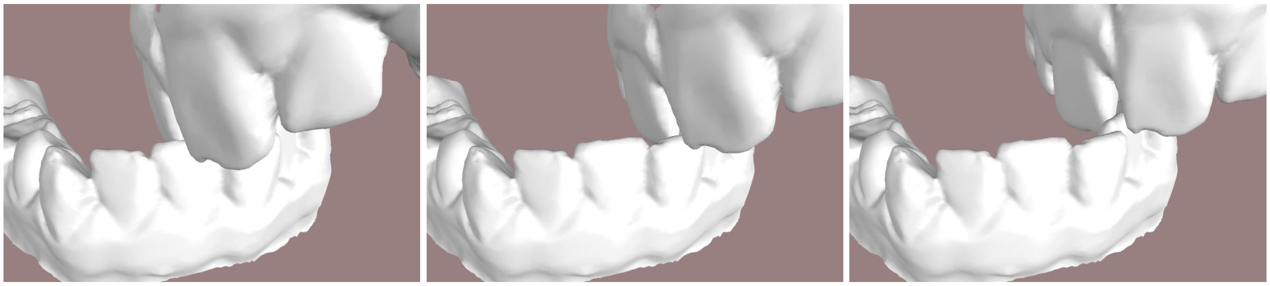

Figure 5: **Dexterous Interaction of Virtual Jaws.** *Three snapshots of an upper jaw (47,339 triangles) being moved over a lower jaw (40,180 triangles), with intricate teeth interaction.*

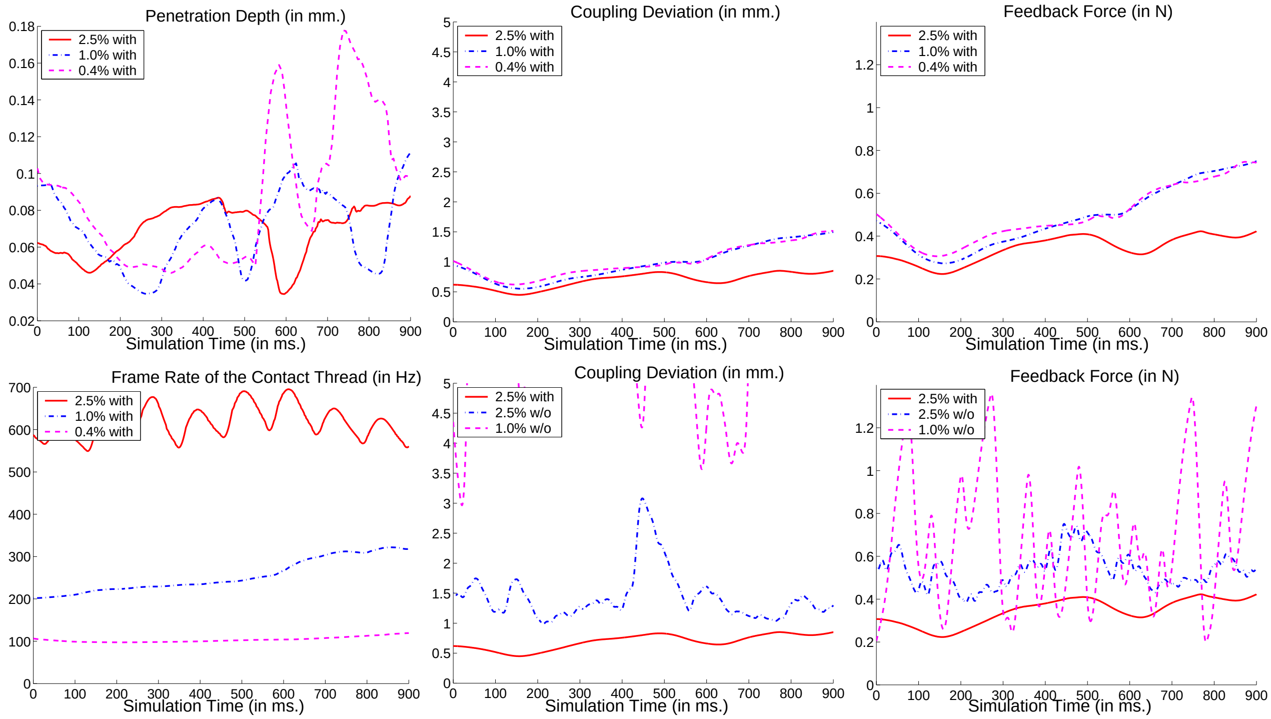

Figure 6: **Analysis of the Linearized Contact Model.** *Comparison of maximum local penetration depth (top left), frame rate of the contact thread (bottom left), coupling deviation (center), and feedback or coupling force (right) using different error tolerances for sensation-preserving simplification, with and without (w/o) linearized contact model.*

our approach with some previous techniques, discuss its limitations, and suggest future research directions.

8.1 Discussion and Limitations

Previous techniques have integrated virtual coupling with penalty-based simulation using explicit integration methods [17]. As we have shown, implicit integration enables a wider range of stiffness and mass values, producing more stable and responsive interaction. Our approach bears some similarities with Wan and McNeely's [25] quasi-static approximation, in the sense that we linearize coupling and contact forces. However, we linearize the forces in the full state space of a rigid body (i.e., position and velocities) and take into account elastic and viscous forces. Our formulation enables the simulation of viscous and inertial effects, which the quasi-static approximations do not compute. The key benefit of quasi-static approximation is that it enables very responsive manipulation with high stability. And, we have shown in this paper that such effects can be achieved using implicit integration as well. Moreover, a viscoelastic virtual coupling adds beneficial filtering effects when contact discontinuities take place.

Contact discontinuities are some of the well known issues associated with penalty-based simulation methods. Contact clustering and virtual coupling enable stable behavior in most situations, as demonstrated by our results; but torque discontinuities can arise, for example, when large flat parallel surfaces are in contact. This situation is a common problem in penalty-based simulation with fixed time steps. Other limitations of our approach include the lack of friction forces in the current implementation and the assumption of a static workspace environment. Friction forces can easily be added using localized friction models

[13]. Our formulation can be easily extended to dynamic environments, computing penalty-based interaction among multiple moving objects, but the cost of the simulation will grow considerably for complex scenes.

8.2 Future Work

As part of future work, we would like to address some of the limitations of our current approach, extending it to multiple dynamic objects, deformable bodies, textured surfaces, and other types of model representations. Our work will benefit from studies of human factors, since the stability and responsiveness of the rendering algorithm can be evaluated from a perceptual perspective. Our work will also benefit from its integration with practical applications, such as training simulators for endoscopic surgery, to help us identify future research needs.

Acknowledgments

This research is supported in part by a UNC Computer Science Alumni Fellowship, National Science Foundation, Office of Naval Research, U.S. Army Research Office, and Intel Corporation. We would like to thank the UNC Gamma group and the anonymous reviewers for their feedback on the earlier drafts of this paper.

References

[1] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. *Virtual Reality Annual International Symposium*, pages 203–210, 1995.

[2] R. J. Adams and B. Hannaford. A two-port framework for the design of unconditionally stable haptic interfaces. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[3] D. Baraff and A. Witkin. Large steps in cloth simulation. *Proc. of ACM SIGGRAPH*, pages 43–54, 1998.

[4] P. J. Berkelman. *Tool-Based Haptic Interaction with Dynamic Physical Simulations Using Lorentz Magnetic Levitation*. PhD thesis, Carnegie Mellon University, 1999.

[5] F. P. Brooks, Jr. , M. O. -Young, J. J. Batter, and P. J. Kilpatrick. Project GROPE — Haptic displays for scientific visualization. *Proc. of ACM SIGGRAPH*, pages 177–185, 1990.

[6] B. Chang and J. E. Colgate. Real-time impulse-based simulation of rigid body systems for haptic display. *Proc. of ASME Dynamic Systems and Control Division*, 1997.

[7] J. E. Colgate and G. G. Schenkel. Passivity of a class of sampled-data systems: Application to haptic interfaces. *Proc. of American Control Conference*, 1994.

[8] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995.

[9] D. Constantinescu, S. E. Salcudean, and E. A. Croft. Impulsive forces for haptic rendering of rigid contact. *Proc. of International Symposium on Robotics*, pages 1–6, 2004.

[10] C. Edmond, D. Heskamp, D. Sluis, D. Stredney, G. Wiet, R. Yagel, S. Weghorst, P. Oppenheimer, J. Miller, M. Levin, and L. Rosenberg. ENT endoscopic surgical simulator. *Proc. of Medicine Meets VR*, pages 518–528, 1997.

[11] A. Gregory, A. Mascarenhas, S. Ehmann, M. C. Lin, and D. Manocha. 6-DoF haptic display of polygonal models. *Proc. of IEEE Visualization Conference*, 2000.

[12] B. Hannaford, J. -H. Ryu, and Y. S. Kim. Stable control of haptics. In M. L. McLaughlin, J. P. Hespanha, and G. S. Sukhatme, editors, *Touch in Virtual Environments*, chapter 3, pages 47–70. Prentice Hall PTR, Upper Saddle River, NJ, 2002.

[13] V. Hayward and B Armstrong. A new computational model of friction applied to haptic rendering. *Experimental Robotics VI*, 2000.

[14] V. Hayward, P. Gregorio, O. Astley, S. Greenish, and M. Doyon. Freedom-7: A high fidelity seven axis haptic device with applications to surgical training. *Experimental Robotics*, pages 445–456, 1998.

[15] D. E. Johnson and P. Willemsen. Accelerated haptic rendering of polygonal models through local descent. *Proc. of Haptics Symposium*, 2004.

[16] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Six-degree-of-freedom haptic rendering using incremental and localized computations. *Presence*, 12(3):277–295, 2003.

[17] W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.

[18] B. E. Miller, J. E. Colgate, and R. A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6), pages 712–719, 1990.

[19] D. D. Nelson, D. E. Johnson, and E. Cohen. Haptic rendering of surface-to-surface sculpted model interaction. *Proc. of ASME Dynamic Systems and Control Division*, 1999.

[20] M. A. Otaduy. *6-DoF Haptic Rendering Using Contact Levels of Detail and Haptic Textures*. PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[21] M. A. Otaduy, N. Jain, A. Sud, and M. C. Lin. Haptic display of interaction between textured models. *Proc. of IEEE Visualization*, pages 297–304, 2004.

[22] M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. *Proc. of ACM SIGGRAPH*, pages 543–553, 2003.

[23] D. Ruspini and O. Khatib. A framework for multi-contact multi-body dynamic simulation and haptic display. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[24] S. E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. *Proc. of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1994.

[25] M. Wan and W. A. McNeely. Quasi-static approximation for 6 degrees-of-freedom haptic rendering. *Proc. of IEEE Visualization*, pages 257–262, 2003.