Haptic Rendering of Complex Deformations through Handle-Space Force Linearization

Carlos Garre Miguel A. Otaduy

URJC Madrid*

Abstract

The force-update-rate requirements of transparent rendering of virtual environments are in conflict with the computational cost required for computing complex interactions between deforming objects. In this paper we introduce a novel method for satisfying high force update rates with deformable objects, yet retaining the visual quality of complex deformations and interactions.

The objects that are haptically manipulated may have many degrees of freedom, but haptic interaction is often implemented in practice through low-dimensional force-feedback devices. We exploit the low-dimensional domain of the interaction for devising a novel linear approximation of interaction forces that can be efficiently evaluated at force-update rates. Moreover, our linearized force model is time-implicit, which implies that it accounts for contact constraints and the internal dynamics of deforming objects. In this paper we show examples of haptic interaction in complex situations such as large deformations, collision between deformable objects (with friction), or even self-collision.

Index Terms: I.2.9 [Robotics]—; I.3.7 [Three Dimensional Graphics and Realism]: Virtual Reality—

1 Introduction

In our interaction with the world around us, we often hold an object (a.k.a. *tool*), and perceive the contact between this object and other objects in the world. Some of the popular applications of computational haptics, such as surgical simulation or assembly and maintanance planning, also follow this interaction paradigm. The tool may be an elastic deformable object, such as organ tissue, a rubber part of an assembly, a piece of cloth, or a soft toy, with many degrees of freedom. However, the part of the tool that we grasp (a.k.a. *handle*) can be considered to move almost rigidly, due to the action of the grasp itself. Many force-feedback devices also exhibit low-dimensional end-effectors, which can be used for interacting with a virtual environment through the handle of a deformable virtual tool.

In this paper, we exploit the low-dimensionality of handle-based interaction for devising a haptic rendering algorithm of deformable objects undergoing complex contact. We decompose the problem of haptic rendering into two subproblems: on the one hand, the simulation of contact and deformations for all objects involved in a virtual environment, and on the other hand, the computation of feedback forces to be displayed to the user. Similarly to others before, our solution follows a multirate approach. A slow-computing thread employs a sufficiently accurate model for computing deformations and contact that produce virtual environment simulations of high visual quality (i.e., rich deformations with many degrees of freedom, precise contact without interpenetrations, robust friction handling, etc.). A fast-computing thread employs a simplified interaction model for computing the dynamics of the handle and the feedback forces to be displayed to the user. One major contribution of our work lies in the specific way in which our handle-space interaction model is defined and computed. Another major contribution is the possibility to produce haptic interactions with complex contact scenarios not possible with other techniques.

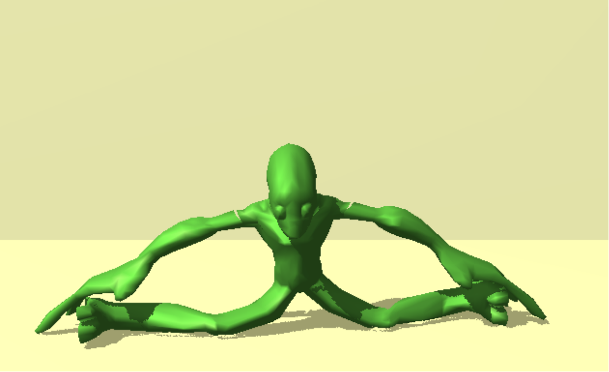


Figure 1: Alien toy being deformed and suffering self-collisions between hands and feet.

Our handle-space force linearization works in two steps. We first solve accurate contact between the tool and the environment, and compute the coupling forces between the handle and the rest of the tool. Then, given a certain contact configuration, we compute a linear approximation of the force acting on the handle, with respect to the state of the handle itself. However, one major feature of this linear approximation is that it implicitly accounts for the contact constraints and the deformation of the tool. We describe our handle-space linearization in detail in Section 3. In a fast thread for feedback force updates, we simply need to evaluate the handle-space linear model given the current state of the handle.

We have evaluated the performance of our handle-space linearization and compared it to standard approaches for coupling a slow-computing simulation to a fast force-update thread. In Section 4 we describe experiments in a simple scenario that served us to validate the model.

But the major benefit of our interaction model is that we can manipulate complex deformable objects in a virtual environment in a dexterous manner, while enjoying a high-quality visual simulation. Section 5 describes the complex scenarios that we tested, which include contact between a deformable tool and other deformable objects in the environment (see Fig. 5), or even self-collision of the tool (as seen in Fig. 1). Our handle-space linearization also allows to naturally hold the tool from different parts, and thereby perceive diverse contact and inertial sensations.

2 Related Work

Rendering of haptic interaction with interesting virtual environments relies to a large extent on the ability to simulate efficiently effects such as deformations and contact between the objects in the virtual environment. We refer the reader to surveys on those topics

*http://www.gmrv.es/Publications/2009/GO09 for more information [13, 23, 20, 34]. The efficiency of the techniques for solving contact and deformation problems is increasingly growing (see [4, 33, 14, 16, 31, 28] for some recent examples), but the complexity of the scenes that can be handled at force update rates is still far from desirable. The quality of the force feedback can be described as the ability to transparently convey contact and inertial forces during manipulation, while guaranteeing stable interaction. The range of contact impedances that can be rendered in a stable manner depends, however, on the force update rate [7], and this sets a major challenge when simulating complex deformations. Measurement-based approaches [19] may decrease the high update-rate requirements, but it is yet unclear how to extend such approaches to model sliding, rolling, or distributed contact between time-varying objects.

The fundamental approach for modeling complex contact is to maximize the efficiency of collision detection, dynamics simulation, and collision response algorithms. In the context of haptic rendering, researchers have proposed solutions for fast collision detection and response between rigid bodies [21, 17, 15], or even collision response between deformable objects [12, 5] (although limited to certain types of deformations). Others have proposed level-of-detail algorithms that select the appropriate object resolution based on contact properties and the available computational time, for rigid bodies [26] or also deformable bodies [5].

Nevertheless, all such techniques may be limited by scene complexity, and unable to deliver high-quality feedback forces at the desirable rates. Those techniques are complementary to our work, since we aim at combining a (possibly slow) visual simulation of the virtual environment with (fast) haptic rendering of the contact and deformation forces. Our work lies closer to earlier approaches that create at run-time a simplified model of contact and deformations, and use this model for feedback force updates.

The simplest way of coupling a slow-update simulation and fast-update force computation is the use of a virtual coupling [8] between the haptic device and a simulation of the handle. Stable rendering can be obtained by designing a passive simulation of the virtual environment and appropriately tuning the coupling parameters. Though stable, the transparency of the rendering may be highly compromised, since the maximum coupling stiffness may be rather low. In a multirate approach, however, the slow-computing simulation also generates an approximate intermediate representation of the interaction with the environment [1], and this intermediate representation is evaluated for force feedback. High rendering stiffness is posible in the multirate approach thanks to the intermediate representation, but rendering quality depends on the quality of the approximations.

Some multirate rendering approaches define active contact constraints in the slow thread and use them to compute contact forces in the fast thread through Signorini's contact model [11], penalty-based methods [27], projection of unconstrained accelerations [24], or a least-squares solution to Poisson's restitution hypothesis for rigid bodies [9]. Others perform approximate local updates of the contact constraints in the fast loop and then apply penalty-based forces [15]. Early approaches to multirate simulation of deformable models considered force extrapolation [30], meshes of different resolution coupled through Norton equivalents [2], or local linearized submeshes for approximating high-frequency behavior [6]. It is also worth noting existing work on stability analysis of multirate rendering algorithms [3].

Our work is most similar to those of Otaduy and Gross [25] and Duriez et al. [11, 12]. Otaduy and Gross linearized the summed contact force and torque acting on a rigid tool w.r.t. other forces and torques (e.g., due to virtual coupling with the device). In contrast, the method presented here captures a (locally) linear representation of the contact between a deformable tool and other deformable objects, as well as the internal force between the deformable part of the tool and a rigid handle. Our method is qualitatively distinct, as it allows the user to manipulate a deformable tool. The approach followed for obtaining the linear representation is similar to the one by Otaduy and Gross, but accounts for much more complex contact between deformable objects. Duriez et al. identify active constraints in the visual loop, and then solve a constrained problem with equality constraints in the haptic loop. Contact forces are transported to the rigid part of the tool, which is connected to the haptic device through virtual coupling. Their approach relies on quasi-rigid deformations, hence it cannot support large deformations, and suffers an increase of the computational cost (in the haptic loop) as the number of contacts grows.

Last, let us point out that local linear force models together with virtual coupling are also used in the context of master-slave robot teleoperation (see [18, 29] for some examples), although in that case the linear force model is obtained through physical contact stiffness estimation.

3 Handle-Space Force Linearization

In this section we describe the details of our handle-space linearization of the contact forces between the tool and the rest of the environment. We start with an outline of the multirate rendering algorithm. Then we describe the constrained dynamics formulation that we use for the visual simulation. Based on this formulation, we present the exact definition of the linearized force model, we outline the algorithm for its run-time computation, and we discuss some implementation details.

3.1 Haptic Rendering Algorithm

As introduced already, we employ a multirate rendering algorithm (schematically depicted in Fig. 2). In a *visual loop*, we execute a sufficiently accurate simulation of the full virtual environment, including contact between the deformable tool and other objects in the environment. In this visual simulation, we drive the motion of the tool handle through a one-way viscoelastic link to the position and orientation of the haptic device. In our implementation, the tool handle is a rigid body, and is coupled to the rest of the handle through internal elastic forces (see $\mathbf{F}_c$ in Fig. 2). The contact between the tool and the environment is transmitted to the handle through these internal forces. Note again that the handle and the haptic device are linked through a one-way coupling, and the handle does not transmit forces to the haptic device.

In a *haptic loop*, we execute a simulation of a proxy of the handle. This proxy is connected through a bidirectional viscoelastic link to the haptic device, thus the actual force feedback perceived by the user is determined by the proxy. The key novelty in our work is the linearized version of the handle-tool coupling, $\mathbf{F}_c^*$. This force mimics the effect produced by the actual coupling between handle and tool, and is aware of the current dynamic configuration of the tool and the existence of contact between the tool and the rest of the environment.

3.2 Dynamics Model of the Virtual Environment

The purpose of our work was not to introduce a new dynamics or contact model. For the visual loop simulation, we have opted for the constrained dynamics model of Otaduy et al. [28], as it provides a robust framework for modeling accurate contact and high-quality visual simulation. We now briefly summarize the application of their method in our setting.

Let us denote with $\mathbf{x}_d$, $\mathbf{x}_h$, $\mathbf{x}_t$, and $\mathbf{x}_e$ the generalized coordinates of the haptic device (d), the handle (h), the tool (t), and the rest of the objects in the virtual environment (e). Similarly, we define the velocities as $\mathbf{v}$ and mass matrices as $\mathbf{M}$, with the corresponding subindices. The forces acting on the handle can be grouped in a vector $\mathbf{F}_h$, and depend on the state of the handle, the haptic device, and the tool. Similarly, we define forces acting on the rest of the

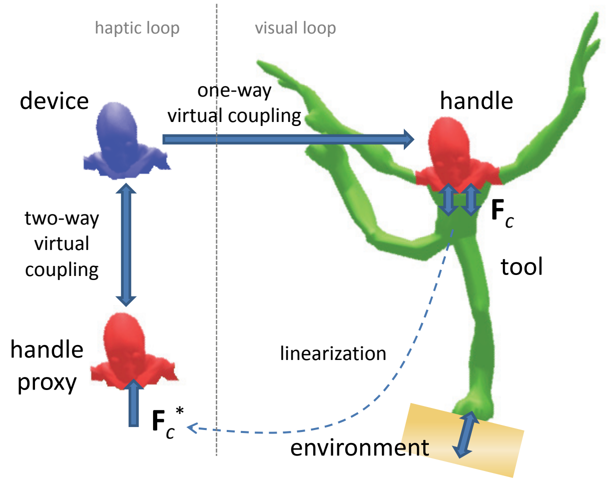


Figure 2: Diagram that depicts the flow of forces in our multirate haptic rendering algorithm, and the concept of handle-space linearization of tool and environment effects. In each simulation step in the visual loop, we compute a linear approximation $\mathbf{F}_c^*$ of the coupling force $\mathbf{F}_c$ between tool and handle, that encapsulates the constrained dynamics of the tool.

tool and the environment. We do not consider forces between the tool and the environment other than contact forces.

In our implementation, we have opted for a co-rotational FE model [22] for simulating the dynamic deformations of the tool and other objects in the environment, and a rigid body model for the handle (see [32] for an extensive treatment of rigid body simulation). We model contact constraints using algebraic inequalities $\mathbf{c} \geq \mathbf{0}$ on the positions of contact points, together with complementarity relationships, and friction through a linearized Coulomb model, all as described in [28]. The action of contact constraints on the generalized coordinates of the system can be modeled through Lagrange multipliers as $\mathbf{J}^T\lambda$, with $\mathbf{J}$ the constraint Jacobian. Altogether, the constrained dynamics formulation can be written as:

$$\begin{aligned} \mathbf{M}_h\dot{\mathbf{v}}_h &= \mathbf{F}_h(\mathbf{x}_h,\mathbf{x}_t,\mathbf{x}_d), \\ \mathbf{M}_t\dot{\mathbf{v}}_t &= \mathbf{F}_t(\mathbf{x}_t,\mathbf{x}_h)+\mathbf{J}_t^T\lambda, \\ \mathbf{M}_e\dot{\mathbf{v}}_e &= \mathbf{F}_e(\mathbf{x}_e)+\mathbf{J}_e^T\lambda, \\ \mathbf{c}(\mathbf{x}_t,\mathbf{x}_e) &\geq \mathbf{0}. \end{aligned} \tag{1}$$

We discretize the dynamic equations using implicit integration (specifically a semi-implicit Backward Euler method), which allows for large time steps. Then, the time-discrete constrained dynamics can be written as:

$$\begin{aligned} \mathbf{A}_h\mathbf{v}_h+\mathbf{A}_{ht}\mathbf{v}_t &= \mathbf{b}_h, \\ \mathbf{A}_{th}\mathbf{v}_h+\mathbf{A}_t\mathbf{v}_t &= \mathbf{b}_t+\mathbf{J}_t^T\lambda, \\ \mathbf{A}_e\mathbf{v}_e &= \mathbf{b}_e+\mathbf{J}_t^e\lambda, \\ \mathbf{J}_t\mathbf{v}_t+\mathbf{J}_e\mathbf{v}_e &\geq \mathbf{b}_\lambda. \end{aligned} \tag{2}$$

Note that the use of implicit integration and the existence of elastic forces between handle and tool effectively produces an inertial coupling between tool and handle in the time-discrete setting. And note also that the implicit position constraints turn into velocity constraints by considering the numerical integration of positions.

Grouping all velocities in one large vector $\mathbf{v} = (\mathbf{v}_h^T\ \mathbf{v}_t^T\ \mathbf{v}_e^T)^T$, the equations above can be rewritten as:

$$\begin{aligned} \mathbf{A}\mathbf{v} &= \mathbf{b}+\mathbf{J}^T\lambda, \\ \mathbf{J}\mathbf{v} &\geq \mathbf{b}_\lambda. \end{aligned} \tag{3}$$

The complete system of equations, including complementarity conditions for non-penetration and friction, is a nonlinear complementarity problem [10], and its solution (following [28]) yields the Lagrange multipliers λ and velocities $\mathbf{v}$.

3.3 Definition of the Linearized Coupling Force

The coupling force between tool and handle can be expressed in general terms as a function of the state of both the tool and the handle, i.e.,

$$\mathbf{F}_c = \mathbf{f}(\mathbf{x}_h,\mathbf{v}_h,\mathbf{x}_t,\mathbf{v}_t). \tag{4}$$

In our implicit time discretization, this force depends on "future" state values. Then, if we account for the integration of positions (in this case, with backward Euler), the coupling force can be expressed as a function:

$$\mathbf{F}_c = \mathbf{f}(\mathbf{x}_{h,0}+\Delta t\mathbf{v}_h,\mathbf{v}_h,\mathbf{x}_{t,0}+\Delta t\mathbf{v}_t,\mathbf{v}_t). \tag{5}$$

Reordering terms, we have

$$\mathbf{F}_c = \mathbf{g}(\mathbf{x}_{h,0},\mathbf{v}_h,\mathbf{x}_{t,0},\mathbf{v}_t). \tag{6}$$

We can conceptually define a time-varying function $(\mathbf{v}_h,\mathbf{v}_t) = \phi(\mathbf{x}_{h,0},\mathbf{v}_{h,0})$ that encapsulates the complete effect of integrating the constrained dynamics formulation, and relates the input state of the handle to the output velocities of the handle and the tool. Let us call ϕ the *discrete update function*. Given the definition of this function, we have all the necessary ingredients for defining our handle-space linear approximation of the coupling force:

$$\mathbf{F}_c^*(\mathbf{x}_h,\mathbf{v}_h) = \mathbf{F}_c(\mathbf{x}_0,\mathbf{v}_0)+\frac{\partial\mathbf{F}_\mathbf{c}}{\partial\mathbf{x}_{h,0}}(\mathbf{x}_h-\mathbf{x}_0)+\frac{\partial\mathbf{F}_c}{\partial\mathbf{v}_{h,0}}(\mathbf{v}_h-\mathbf{v}_0). \tag{7}$$

The various terms of the linear model are computed in the visual loop, for position and velocity $(\mathbf{x}_0,\mathbf{v}_0)$ of the handle. The model is evaluated in the haptic loop, for position and velocity $(\mathbf{x}_h,\mathbf{v}_h)$ of the proxy handle. The linear model acts on the proxy handle as an approximation of the combined system tool-environment.

In each step of the visual simulation we must compute the derivative terms in Eq. (7). They can be obtained by applying the chain rule and the implicit definition of the force in Eq. (6):

$$\begin{aligned} \frac{\partial\mathbf{F}}{\partial\mathbf{x}_{h,0}} &= \frac{\partial\mathbf{g}}{\partial\mathbf{x}_{h,0}}+\frac{\partial\mathbf{g}}{\partial\mathbf{v}_h}\frac{\partial\mathbf{v}_h}{\partial\mathbf{x}_{h,0}}+\frac{\partial\mathbf{g}}{\partial\mathbf{v}_h}\frac{\partial\mathbf{v}_t}{\partial\mathbf{x}_{h,0}}, \\ \frac{\partial\mathbf{F}}{\partial\mathbf{v}_{h,0}} &= \frac{\partial\mathbf{g}}{\partial\mathbf{v}_h}\frac{\partial\mathbf{v}_h}{\partial\mathbf{v}_{h,0}}+\frac{\partial\mathbf{g}}{\partial\mathbf{v}_t}\frac{\partial\mathbf{v}_h}{\partial\mathbf{v}_{h,0}}. \end{aligned} \tag{8}$$

The terms $(\frac{\partial\mathbf{g}}{\partial\mathbf{x}_{h,0}},\frac{\partial\mathbf{g}}{\partial\mathbf{v}_h},\frac{\partial\mathbf{g}}{\partial\mathbf{v}_t})$ in Eq. (8) are given by the known force expression in Eq. (6). The complexity in the definition of the linearized coupling force comes from the terms $(\frac{\partial\mathbf{v}_h}{\partial\mathbf{x}_{h,0}},\frac{\partial\mathbf{v}_h}{\partial\mathbf{v}_{h,0}},\frac{\partial\mathbf{v}_t}{\partial\mathbf{x}_{h,0}},\frac{\partial\mathbf{v}_t}{\partial\mathbf{v}_{h,0}})$, which are nothing else but the derivatives of the discrete update function ϕ. Next, we describe the computation of these derivatives.

3.4 Computation of the Linearized Coupling Force

Let us consider a certain time step in the visual simulation loop. Once the constrained dynamics are solved, we can identify the active contact constraints and discard inactive constraints, turning the inequality constraints in Eq. (2) into equality constraints. We proceed similarly with sticking and slipping friction constraints. For

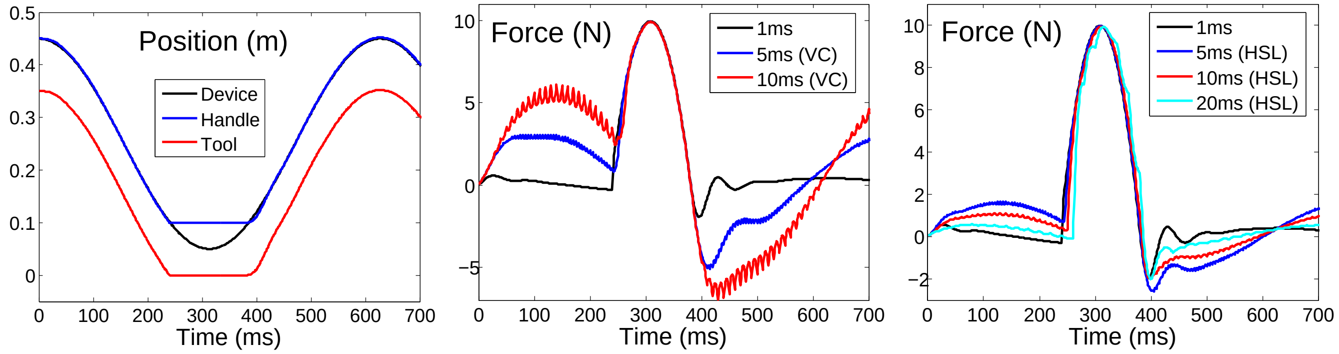


Figure 3: Evaluation of 1-Dimensional Handle-Space Linearization. Left: Positions of device, handle, and tool; Middle: Output force for virtual coupling (VC) with various time-step sizes; Right: Output force for handle-space linearization (HSL) with various time-step sizes.

simplicity, let us assume that all constraints are active, hence we retain the notation from Eq. (2), but with equality constraints. The Lagrange multipliers and constrained velocities can then be formulated as:

$$\lambda = \mathbf{A}_\lambda^{-1}(\mathbf{b}_\lambda - \mathbf{J}\mathbf{A}^{-1}\mathbf{b}), \tag{9}$$

$$\mathbf{v} = \mathbf{A}^{-1}(\mathbf{b} + \mathbf{J}^T\mathbf{A}_\lambda^{-1}(\mathbf{b}_\lambda - \mathbf{J}\mathbf{A}^{-1}\mathbf{b})), \tag{10}$$

$$\text{with } \mathbf{A}_\lambda = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T.$$

Recall that, for the computation of the linearized coupling force as in Eq. (8), we seek the derivatives of the velocities $\mathbf{v}$ w.r.t. the state of the handle. We will assume that the inertia, stiffness, and constraints do not change significantly between updates, therefore the only terms that depend on the state of the handle in Eq. (10) are the right-hand sides $(\mathbf{b}, \mathbf{b}_\lambda)$. We can then formulate the derivative w.r.t. the initial position of the handle (and similarly for the initial velocity):

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}_{h,0}} = \mathbf{A}^{-1}\left(\frac{\partial \mathbf{b}}{\partial \mathbf{x}_{h,0}} + \mathbf{J}^T\mathbf{A}_\lambda^{-1}\left(\frac{\partial \mathbf{b}_\lambda}{\partial \mathbf{x}_{h,0}} - \mathbf{J}\mathbf{A}^{-1}\frac{\partial \mathbf{b}}{\partial \mathbf{x}_{h,0}}\right)\right). \tag{11}$$

The right-hand sides $(\mathbf{b}, \mathbf{b}_\lambda)$ are given by the contact constraints and the forces acting on the handle and the tool, hence their derivatives will be obtained by differentiating the known implementations in each case.

At this point, we can define an algorithm for the computation of the linearized coupling model. Every time step in the visual loop we do:

1. Solve for the constrained dynamics problem in Eq. (2).
2. Determine active constraints and reformulate the linear system in Eq. (2) using equality constraints.
3. Compute the derivatives of right-hand sides $(\mathbf{b}, \mathbf{b}_\lambda)$ w.r.t. the state of the handle.
4. Compute the derivatives of handle and tool velocities w.r.t. the state of the handle using Eq. (11).
5. Compute the derivatives of the implicit coupling force in Eq. (6) w.r.t. the handle and tool velocities.
6. Put all terms together and compute the linearized model as defined in Eq. (7) and Eq. (8).

3.5 Implementation

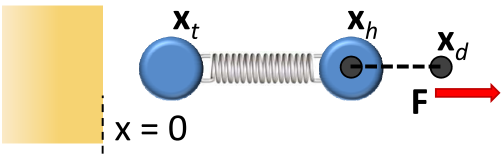
As mentioned earlier, we have modeled the handle as a rigid body. In that case, the derivatives of right-hand sides $(\mathbf{b}, \mathbf{b}_\lambda)$ are matrices with only a few columns (6 for the derivative w.r.t. velocity, and 7 for a derivative w.r.t. position, with a quaternion implementation of rotations).

The main cost in the algorithm above is the solution of systems $\mathbf{y} = \mathbf{A}_\lambda^{-1}\mathbf{z}$ in Eq. (11). In the general case, $\mathbf{A}_\lambda$ is dense and large. In our implementation, however, we have approximated this matrix by taking only its diagonal terms, which amounts to considering different contacts to be inertially decoupled. Such decoupling has yet another additional benefit, as the linearized model depends only on the tool and the handle, but not on other objects in the virtual environment. Effectively, we may drop the terms and equations for $\mathbf{v}_e$ from Eq. (2). This approximation may not be applicable if the objects in the environment are moving fast, and we are exploring solutions for that situation. Note also that we only diagonalize $\mathbf{A}_\lambda$ in the computation of the linearized coupling model, *not* in the actual solution of constrained dynamics in the visual loop.

Solving a linear system $\mathbf{y} = \mathbf{A}^{-1}\mathbf{z}$ in Eq. (11) corresponds simply to doing one solve of forward dynamics. Considering the handle to be a rigid body, and assuming the inertial decoupling of contacts described above, we need to solve the forward dynamics for the tool and the handle $2 \times (6+7) = 26$ times in total. We speed-up these solves by pre-computing a Cholesky factorization of $\mathbf{A}_t$ once every visual simulation step (with the TAUCS library [35]).

4 Model Evaluation

We have designed simple 1-dimensional experiments (in Matlab) for evaluating the quality of the handle-space linearization. Specifically, we have modeled the following setup:



where a point modeling the haptic device $(\mathbf{x}_d)$ drags through a viscoelastic coupling the handle of a simulated tool. The tool contains two point masses, one of them modeling the handle $(\mathbf{x}_h)$ and another modeling the rest of the tool $(\mathbf{x}_t)$, linked through a spring of length $L = 0.1$m that models the elastic properties of the tool. The tool may collide against a virtual wall located at $\mathbf{x} = 0$.

The position of the haptic device is prescribed. It starts at $x = 0.45$ and moves left until $x = 0.05$, to later move right again (as shown in Fig. 3-left). When it moves left, the tool should collide with the wall. This simulation models the behavior of our visual simulation loop as described in Section 3.1. We also execute a haptic loop, that implements the haptic proxy and the linearized coupling force (not shown in the figure). In this one-dimensional

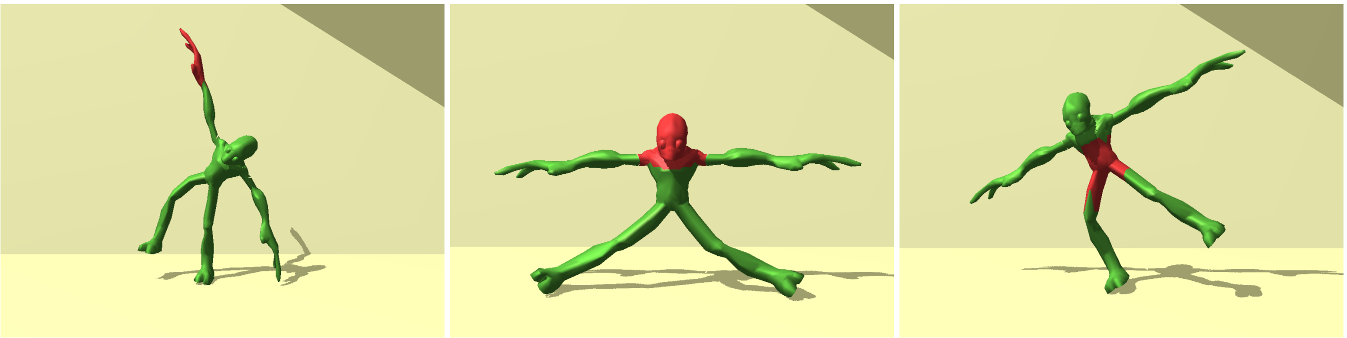


Figure 4: Alien toy being manipulated through three different handles (painted in red). In the rightmost screenshot, notice the deformation at the leg produced by frictional contact with the ground.

example, the coupling force is actually the force of the spring that links the handle and the tool, and its linearization will account for the dynamics of the tool subject to the contact with the wall.

We have executed two types of experiments, one type mimicking our haptic rendering algorithm using handle-space linearization (HSL), and another type that renders forces to the haptic device through a virtual coupling (VC) with the handle. We have executed the haptic loop in all cases with a time step of 1 ms, and the simulation loop with time steps varying from 1 ms to 20 ms. Fig. 3-left shows the evolution of the position of the device, the handle, and the tool. Such positions correspond to what the user would *see* in a 3D simulation. The positions varied little across experiments.

However, the feedback force to be sent to the haptic device was much more sensitive to the time step of the visual loop and the interaction method. Fig. 3-middle compares feedback forces obtained with virtual coupling to a reference visual simulation executed with a 1 ms time step. At a visual-loop time step of 10 ms, the perceived viscosity of the virtual coupling is almost as large as the contact force, thereby masking out the perception of contact. Fig. 3-right compares instead feedback forces with our handle-space linearization, showing much better behavior even for a time step of 20 ms.

5 Complex Experiments

We have executed our 3D simulations on a quad-core 2.4 GHz PC with 3 GB of memory (although we have only used two of the processors, for the visual and haptic loops). We manipulated the models using a Phantom Omni haptic device from SensAble Technologies. Videos available in the project webpage show real-time screen captures of all the examples, highlighting the high visual quality of the contact.

One of our examples consisted of an alien toy being manipulated and deformed. The triangle surface of the model is embedded on a tetrahedral mesh that we use for the dynamic simulation of deformations. In the case of the alien, this mesh has 288 tetrahedra. Despite using this mesh for computing deformations, we detect and handle collisions on the triangle surface, by mapping contact forces back to FEM mesh nodes. The visual simulation loop was able to maintain a frame rate of approximately 30 Hz on average. Fig. 1 highlights the fact that our technique can handle large deformations and even self-collisions of the tool. The inertial effects produced by the self-collisions are then mapped to the handle through the coupling force. Fig. 4 shows screenshots of the simulation when placing the handle in different parts of the alien toy, and the deformation effects produced by gravity and friction.

The other example consisted of two soft bunnies in contact, with one bunny model manipulated with a haptic device. Fig. 5 shows several screenshots of the simulation. Similar to the alien-toy example, we embedded the triangle surfaces in low-resolution tetrahedral meshes for the purpose of computing dynamic deformations, but detected and solved contact on the triangle surfaces themselves. Previous methods for haptically interacting with deformable objects were not able to handle large deformations and detailed contact simultaneously, while our method does this satisfactorily.

6 Discussion and Future Work

In this paper, we have presented a haptic rendering algorithm for manipulating deformable objects in contact with other deformable objects in the environment. Our algorithm is founded on a multirate architecture and a novel intermediate representation that leverages the low-dimensional domain of the handle's motion. Specifically, in each step of the visual loop simulation we compute a linear approximation of the coupling force between the handle and the rest of the tool, and later evaluate this linear approximation in each step of the haptic loop. The major feature of our handle-space linearization is that it accounts for the constrained dynamics of the tool.

While designing our haptic rendering algorithm, we have faced several alternatives in the specifics of the linearization. For example, one could linearize in the state-space, as done here, or in the space of forces, as done in [25], which can be interpreted as a Jacobian that relates contact forces to manipulator forces. It is not yet fully clear to us which option is better and in which situation, and a thorough analysis of the linearization alternatives would be worthwhile. The type of linearization may also have an impact on its computational cost, on the perceived viscosity, or even on possible active behavior due to sudden changes in the linearization. We are still at early stages in this analysis.

Our current formulation does not consider direct contact on the handle, but this could be integrated by also linearizing the contact forces. Furthermore, for the purpose of computing the linearization we have approximated the solution of contact constraints by decoupling the contacts, and we are investigating more accurate approximations with little additional cost.

The work that we have presented here is orthogonal to ongoing research on the improvement of collision detection, collision response, and dynamic deformations, and we believe that they could benefit from each other. In our experience, a slow update of the linearized forces may lead to an increase in the perceived viscosity.

Last, we are evaluating the inclusion of our rendering algorithm in applications that require contact between deformable tools and the environment, such as soft-tissue manipulation in the context of biomedical simulation.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, Andri Bühler, Luismi Serrano, Susana Mata and the rest of the GMRV group at URJC. This work was funded in part by the URJC - Comunidad de Madrid project CCG08-URJC/DPI-3647.

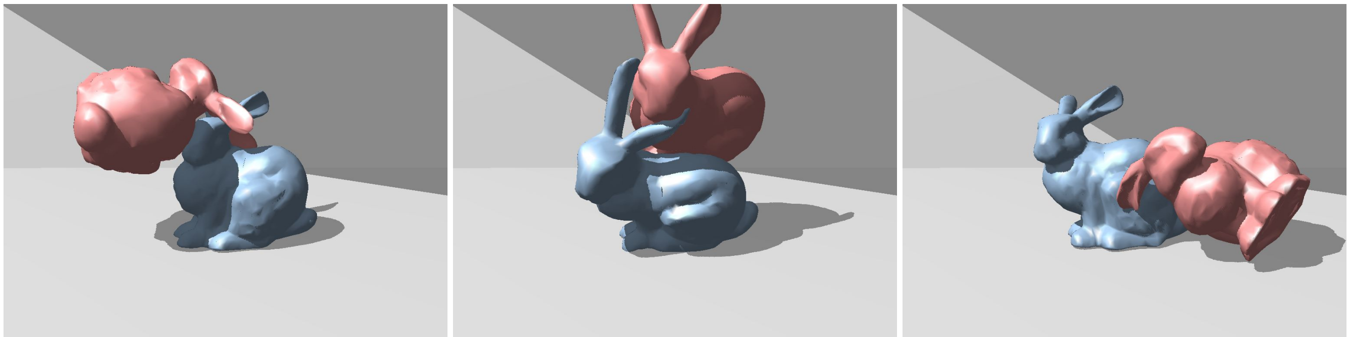


Figure 5: Two soft bunnies in contact. Our haptic rendering algorithm supports large and local deformations, and distributed and detailed contact, all simultaneously.

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] O. R. Astley and V. Hayward. Multirate haptic simulation achieved by coupling finite element meshes through norton equivalents. *Proc. of IEEE International Conference on Robotics and Automation*, 1998.

[3] F. Barbagli, D. Prattichizzo, and K. J. Salisbury. Dynamic local models for stable multi-contact haptic interaction with deformable objects. *Proc. of Haptics Symposium*, 2003.

[4] J. Barbič and D. L. James. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *Proc. of ACM SIGGRAPH*, 2005.

[5] J. Barbič and D. L. James. Six-DoF haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Trans. on Haptics*, 1(1), 2008.

[6] M. C. Çavuşoğlu and F. Tendick. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. *Proc. of IEEE International Conference on Robotics and Automation*, pages 2458–2465, 2000.

[7] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. *IEEE International Conference on Robotics and Automation*, pages 3205–3210, 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*, pages pp. 140–145, 1995.

[9] D. Constantinescu, S. E. Salcudean, and E. A. Croft. Local model of interaction for realistic manipulation of rigid virtual worlds. *International Journal of Robotics Research*, 24(10), 2005.

[10] R. W. Cottle, J. S. pang, and R. E. Stone. The linear complementarity problem. *Academic-Press, Inc.*, 1992.

[11] C. Duriez, C. Andriot, and A. Kheddar. Signorini's contact model for deformable objects in haptic simulations. *Proc. of IEEE/RSJ IROS*, 2004.

[12] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *Proc. of IEEE TVCG*, 12(1), 2006.

[13] S. F. Gibson and B. V. Mirtich. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratory, 1997.

[14] D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun. Robust treatment of simultaneous collisions. *Proc. of ACM SIGGRAPH*, 2008.

[15] D. E. Johnson, P. Willemsen, and E. Cohen. 6-DOF haptic rendering using spatialized normal cone search. *IEEE TVCG*, 11(6), 2005.

[16] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai. Staggered projections for frictional contact in multibody systems. *Proc. of ACM SIGGRAPH Asia*, 2008.

[17] 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.

[18] C.-P. Kuan and K.-Y. Young. Vr-based teleoperation for robot compliance control. *Journal of Intelligent and Robotic Systems*, 30(4):377–398, 2001.

[19] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE TVCG*, 12(2), 2006.

[20] M. C. Lin and D. Manocha. Collision and proximity queries. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 2003.

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

[22] M. Müller and M. Gross. Interactive virtual materials. *Proc. of Graphics Interface*, 2004.

[23] A. Nealen, M. Müller, R. Keiser, E. Boxermann, and M. Carlson. Physically based deformable models in computer graphics. *Eurographics STAR*, 2005.

[24] M. Ortega, S. Redon, and S. Coquillart. A Six-Degree-of-Freedom God-Object method for haptic display of rigid bodies with surface properties. *IEEE TVCG*, 13(3), 2007.

[25] M. A. Otaduy and M. Gross. Transparent rendering of tool contact with compliant environments. *Proc. of World Haptics Conference*, 2007.

[26] M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, pages 543–553, 2003.

[27] M. A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-DoF manipulation. *IEEE Trans. on Robotics*, 22(4), 2006.

[28] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. Implicit contact handling for deformable objects. *Proc. of Eurographics*, 2009.

[29] J. Park and O. Khatib. A haptic teleoperation approach based on contact force control. *International Journal of Robotics Research*, 25(5-6):575–591, 2006.

[30] G. Picinbono, J.-C. Lombardo, H. Delingette, and N. Ayache. Anisotropic elasticity and force extrapolation to improve realism of surgery simulation. *Proc. of ICRA*, 2000.

[31] G. Saupin, C. Duriez, and S. Cotin. Contact model for haptic medical simulation. *Proc. of International Symposium on Computational Models for Biomedical Simulation*, 2008.

[32] A. A. Shabana. *Dynamics of Multibody Systems*. John Wiley and Sons, 1989.

[33] J. Spillmann, M. Becker, and M. Teschner. Non-iterative computation of contact forces for deformable objects. *Journal of WSCG*, 2007.

[34] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Furhmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. *Computer Graphics Forum*, 24(1), 2005.

[35] S. Toledo, D. Chen, and V. Rotkin. Taucs: A library for sparse linear solvers. 2003.