Transparent Rendering of Tool Contact with Compliant Environments

Miguel A. Otaduy   Markus Gross
Computer Graphics Laboratory, ETH Zurich
{otaduy,grossm}@inf.ethz.ch

Abstract

Transparent haptic rendering of the contact between a tool and its environment requires very frequent update of the contact forces acting on the tool. Given a rigid tool and a deformable environment, we define contact constraints by solving a constrained dynamic simulation problem, typically at a low update rate. A generalized contact Jacobian defines velocities at the constraints given the velocity of the rigid tool. We define an inverse of the contact Jacobian that is dynamically consistent with the constraints and, once we know other forces acting on the rigid tool, it allows a fast update of the accumulated contact forces, and thereby highly transparent rendering.

1 Introduction

Haptic rendering of the manipulation of a tool and its interaction with the environment provides tactile presence in applications such as surgical training [8] or maintenance planning [16]. While the objects in the environment may present diverse mechanical behavior, in many applications the tool can be considered to be rigid. Therefore, it is of special relevance to design effective haptic rendering algorithms for the interaction between a rigid tool and a compliant environment.

Similar to many existing approaches, we adopt a multirate architecture for enhancing rendering transparency. On a slow thread, we execute a constrained dynamic simulation of the interaction between the rigid tool and a deformable environment. At the same time, we identify the parameters of a simplified contact model which will be used for a fast update of the dynamics of the tool, and thus for synthesizing the forces to be rendered by the haptic device.

The major contribution of our paper lies on the design of a contact model that, knowing other forces acting on the rigid tool, permits an extremely fast linear approximation of the accumulated effect of contact constraint forces. Our contact model is based on an extension of the concept of contact Jacobian and its dynamically consistent inverse [14]. The use of the linear contact model, together with implicit integration of the dynamics of the tool, enable highly transparent (while stable) rendering of contact during interaction with complex deformable objects, as shown in several examples.

In the next section, we discuss previous approaches for haptic rendering of object-object contact, focusing on their methods for facilitating multirate algorithms. In Section 3, we overview our multirate rendering algorithm. Section 4 describes the collision-free dynamics of a rigid tool and deformable objects, the addition of a virtual coupling for stable haptic manipulation, and implicit integration of the equations of motion. Section 5 presents our contact handling approach, with a constraint-based solution for the slow thread, and our novel contact model for the fast thread. We describe experiments and results in Section 6, and discuss future work in Section 7.

2 Related Work

Haptic rendering of the interaction between a tool and a deformable environment has been extensively investigated in the last decade, leading to a variety of approaches that differ from each other in the methods employed for modeling elastic deformations, detecting collisions, or computing contact forces. But the goal of all rendering techniques is to transparently convey contact 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 [5], and this sets a major challenge when simulating compliant environments.

By using a virtual coupling [6] to connect the haptic device to a simulation of the tool, 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 at low update rates, since the maximum coupling stiffness may be rather low. Independently of the simulation and collision detection methods employed, and the mechanical characteristics of the tool or the environment, a common solution for enhancing the transparency of haptic rendering is to devise a multirate algorithm. A slow process computes accurate interaction between the tool and the environment, and updates an approximate but simple intermediate representation [1]. A fast process synthesizes the forces to be sent to the device, using the intermediate representation.

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 [9], penalty-based methods [19], projection of unconstrained accelerations [18], or a least-squares solution to Poisson's restitution hypothesis for rigid bodies [7]. Others perform approximate local updates of the contact constraints in the fast loop and then apply penalty-based forces [13]. It is also worth noting existing work on stability analysis of multirate rendering algorithms [2].

We adopt continuum mechanics methods to simulate deformable objects, as they allow physically-based modeling of mechanical properties. Some of the methods that have been used for haptic rendering or other interactive applications include linear FEM with matrix condensation [3], corotational linear FEM for stable large deformations [17], the boundary element method and use of capacitance matrices [12], or quasi non-linear elasticity with precomputation of response functions [8].

For contact handling, penalty methods are easy to implement but rely on existence of interpenetration and suffer from loss of passivity, although recent local models alleviate the latter problem [15]. Constraint-based methods, which have a higher computational cost but handle non-penetration accurately, have been applied in a variety of flavours. Cotin et al. [8] applied equality position constraints at contacts, and solved for contact forces using Lagrange multipliers. Duriez et al. [10] eliminated sticking problems by adopting Signorini's contact model, and formulating a linear complementarity problem (LCP) that extensively exploited precomputation of linear contact responses. In order to support both deformable and rigid environment objects, we compute collision response through velocity constraints [11]. Our current formulation is limited to equality constraints without friction, but we expect that it can be extended to inequality constraints.

3 Overview of the Rendering Algorithm

We adopt a multirate rendering algorithm, and we create two instances of the rigid tool manipulated by the user, as depicted in Figure 1. The *visual thread*, typically running at a low update rate (as low as tens of Hz), performs a full simulation of the *visual tool* coupled to the haptic device and interacting with a deformable environment. For simplicity, from now on we will assume that the environment consists of one deformable body. The *haptic thread*, running at 1 kHz in our system, performs the simulation of the *haptic tool* and computes force values to be rendered by the haptic device. Collision detection and full constraint-based collision response are only computed in the visual thread. At the same time, we update the parameters of the linear contact model defined by the dynamically consistent inverse of the contact Jacobian. This linear model (described in detail in Section 5.3) can be evaluated with a fixed, low number of operations, and ensures extremely fast update of contact forces in the haptic thread.

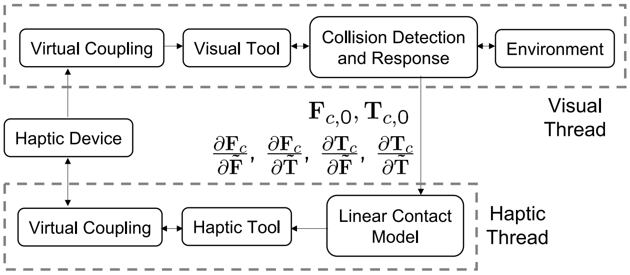
Virtual Coupling
Visual Tool
Collision Detection and Response
Environment
Visual Thread
$\mathbf{F}_{c,0}, \mathbf{T}_{c,0}$
$\frac{\partial \mathbf{F}_c}{\partial \tilde{\mathbf{F}}}, \frac{\partial \mathbf{F}_c}{\partial \tilde{\mathbf{T}}}, \frac{\partial \mathbf{T}_c}{\partial \tilde{\mathbf{F}}}, \frac{\partial \mathbf{T}_c}{\partial \tilde{\mathbf{T}}}$
Haptic Device
Virtual Coupling
Haptic Tool
Linear Contact Model
Haptic Thread

Figure 1. Multirate Rendering Architecture.

Every frame of the visual thread, we:

1. Apply the coupling force to the visual tool.
2. Solve a collision-free update of the visual tool (See Section 4.1).
3. Solve a collision-free update of the deformable body (See Section 4.2).
4. Execute collision detection and identify contact constraints (See Section 5.1).
5. Solve for contact forces and post-collision state (See Section 5.2).
6. Compute the parameters of the linear contact model (See Section 5.3).
7. Perform sanity-check collision detection and correct penetrating points of the deformable body.

Every frame of the haptic thread, we:

1. Apply the coupling force to the haptic tool.
2. Solve a collision-free update of the haptic tool.
3. Compute collision response using the linear contact model (See Section 5.3).
4. Compute the post-collision state.
5. Compute coupling force to be output to the device.

4 Contact-Free Dynamics

In this section we describe the formulation of equations of motion for the rigid tool and deformable objects. We also show their time-discretization, and how to decouple the collision-free update from collision response.

4.1 Simulation of the Tool

We define the state of the tool by the position of its center of mass $\mathbf{x}$, the velocity of the center of mass $\mathbf{v}$, a quaternion $\mathbf{q}$ that describes the orientation, and the angular momentum $\mathbf{L}$. Given the mass m of the tool, mass matrix M, angular velocity $\omega = M^{-1}\mathbf{L}$, forces $\mathbf{F} + J_{\mathbf{v}}^T\lambda$ and torque $\mathbf{T} + J_{\omega}^T\lambda$, the dynamics of the tool are governed by the following ODEs:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v}, & \dot{\mathbf{q}} &= \frac{1}{2}\omega_q\mathbf{q} = \Omega\mathbf{q} = Q\omega, \\ m\dot{\mathbf{v}} &= \mathbf{F} + J_{\mathbf{v}}^T\lambda, & \dot{\mathbf{L}} &= \mathbf{T} + J_{\omega}^T\lambda. \end{aligned} \tag{1}$$

The expressions $\Omega\mathbf{q} = Q\omega$ represent quaternion products as matrix-vector multiplications. We separate contact forces $J_{\mathbf{v}}^T\lambda$ from other forces $\mathbf{F}$ (and similarly for the torque). The formulation of contact forces is described in detail in Section 5.1. Given this separation of forces, and as done by others before [11, 10], at every time step we can separate a collision-free update of velocities $\mathbf{v}^-$ and a collision impulse $\delta\mathbf{v}$ (and similarly for the other state variables). Then, and accounting for time-discretization of the motion equations, the velocity and momentum update can be solved by the following linear systems:

$$\begin{aligned} A_{\mathbf{v}}\left(\mathbf{v}^- + \delta\mathbf{v}\right) &= \Delta t\tilde{\mathbf{F}} + \Delta t J_{\mathbf{v}}^T\lambda, \\ A_{\mathbf{L}}\left(\mathbf{L}^- + \delta\mathbf{L}\right) &= \Delta t\tilde{\mathbf{T}} + \Delta t J_{\omega}^T\lambda. \end{aligned} \tag{2}$$

$$\begin{aligned} \mathbf{v}^- &= \Delta t A_{\mathbf{v}}^{-1}\tilde{\mathbf{F}}, & \delta\mathbf{v} &= \Delta t A_{\mathbf{v}}^{-1} J_{\mathbf{v}}^T\lambda, \\ \mathbf{L}^- &= \Delta t A_{\mathbf{L}}^{-1}\tilde{\mathbf{T}}, & \delta\mathbf{L} &= \Delta t A_{\mathbf{L}}^{-1} J_{\omega}^T\lambda. \end{aligned} \tag{3}$$

We have used implicit backward Euler with a first order approximation of forces, as this allows stable simulation with very low mass, and thus highly transparent rendering [19]. In that case, the discrete-time matrices $A_{\mathbf{v}}$ and $A_{\mathbf{L}}$, and the discrete-time force vectors $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{T}}$, can be written as:

$$\begin{aligned} A_{\mathbf{v}} &= mI - \Delta t\frac{\partial\mathbf{F}}{\partial\mathbf{v}} - \Delta t^2\frac{\partial\mathbf{F}}{\partial\mathbf{x}}, \\ \tilde{\mathbf{F}} &= \mathbf{F} + \left(\frac{m}{\Delta t}I - \frac{\partial\mathbf{F}}{\partial\mathbf{v}}\right)\mathbf{v}, \end{aligned}$$

$$\begin{aligned} A_{\mathbf{L}} &= I - \Delta t\frac{\partial\mathbf{T}}{\partial\mathbf{L}} - \Delta t^2\frac{\partial\mathbf{T}}{\partial\mathbf{q}}\tilde{Q}M^{-1}, \\ \tilde{Q} &= \left(I - \Delta t\Omega\right)^{-1}Q, \\ \tilde{\mathbf{T}} &= \mathbf{T} + \left(\frac{1}{\Delta t}I - \frac{\partial\mathbf{T}}{\partial\mathbf{L}}\right)\mathbf{L}. \end{aligned} \tag{4}$$

We would like to note that, in the computation of Jacobians for implicit integration, we do not account for changes of the inertia matrix between time steps. Also, the external forces $\mathbf{F}$ are limited to gravity and a virtual coupling force applied at the center of mass, and the external torques $\mathbf{T}$ are limited to the virtual coupling torque. If all forces are applied at the center of mass, the equations for translation and rotation update are decoupled, as shown above. With arbitrary forces the equations would be coupled, and we would need to solve one linear system with six equations, instead of two systems with three equations each. The rest of the formulation in the paper would have to be accordingly adapted. For the definition of the coupling force and torque and their Jacobians, please refer to [16, 19].

4.2 Simulation of Deformable Objects

We have opted for corotational FEM methods with linear elasticity for modeling deformable objects [17], because they offer a good trade-off between computational efficiency, robustness, and the richness of effects that can be modeled. We would like to note, however, that our formulation accepts any other model that supports local linearization, as we do not rely on precomputations of capacitance matrices, condensation, etc. After FEM discretization, we obtain lumped diagonal mass matrix $M_{\mathbf{u}}$ and damping matrix D, and a warped stiffness matrix K for each time step. Given external forces $\mathbf{F}_{\mathbf{u}}$, contact forces $J_{\mathbf{u}}^T\lambda$, and node displacements $\mathbf{u}$, the dynamic motion equations of the deformable body are:

$$M_{\mathbf{u}}\ddot{\mathbf{u}} = \mathbf{F}_{\mathbf{u}} - K(\mathbf{u} - \mathbf{u}_0) - D\dot{\mathbf{u}} + J_{\mathbf{u}}^T\lambda. \tag{5}$$

Similarly to the rigid tool, we separate the collision-free update of velocities $\dot{\mathbf{u}}^-$ from the collision impulse $\delta\dot{\mathbf{u}}$. After time-discretization of the motion equations, we have:

$$\begin{aligned} A_{\mathbf{u}}\left(\dot{\mathbf{u}}^- + \delta\dot{\mathbf{u}}\right) &= \Delta_t\left(\tilde{\mathbf{F}}_{\mathbf{u}} + J_{\mathbf{u}}^T\lambda\right), \\ \dot{\mathbf{u}}^- = \Delta t A_{\mathbf{u}}^{-1}\tilde{\mathbf{F}}_{\mathbf{u}}, &\quad \delta\dot{\mathbf{u}} = \Delta t A_{\mathbf{u}}^{-1} J_{\mathbf{u}}^T\lambda. \end{aligned} \tag{6}$$

With backward Euler implicit integration, the discrete-time mass matrix $A_{\mathbf{u}}$ and discrete-time force vector $\mathbf{F}_{\mathbf{u}}$ can be written as:

$$\begin{aligned} A_{\mathbf{u}} &= M_{\mathbf{u}} + \Delta t\left(D - \frac{\partial\mathbf{F}_{\mathbf{u}}}{\partial\dot{\mathbf{u}}}\right) + \Delta t^2\left(K - \frac{\partial\mathbf{F}_{\mathbf{u}}}{\partial\mathbf{u}}\right), \\ \tilde{\mathbf{F}}_{\mathbf{u}} &= \mathbf{F}_{\mathbf{u}} + \left(\frac{1}{\Delta t}M_{\mathbf{u}} - \frac{\partial\mathbf{F}_{\mathbf{u}}}{\partial\dot{\mathbf{u}}}\right)\dot{\mathbf{u}} - K\left(\mathbf{u} - \mathbf{u}_0\right). \end{aligned} \tag{7}$$

5 Contact Handling

As outlined in Section 3, we begin every time step of the visual thread by performing a collision-free dynamic update, followed by collision detection and identification of constraints. In this section, we describe the formulation of constraints, the computation of constraint forces and collision response in the visual thread, and the formulation of the linear model to be used in the haptic thread.

5.1 Formulation of Constraints

Given the state of the tool and the deformable body at the end of the previous time step and the state after the collision-free update, we perform a continuous collision detection test [20] and we identify colliding primitives. For each colliding node of the deformable body, we define a constraint based on the earliest contact. Collisions on edges or faces can be transmitted to nodes based on barycentric coordinates [10].

Given a contact for node $\mathbf{u}_i$ at position $\mathbf{p}_i$ and with normal $\mathbf{n}_i$, we define a velocity constraint as follows:

$$\begin{aligned}
&\mathbf{n}_i^T\left(\mathbf{v}+\omega\times(\mathbf{p}_i-\mathbf{x})-\dot{\mathbf{u}}_i\right)=0, \qquad (8)\\
&\mathbf{j}_{\mathbf{v},i}\mathbf{v}+\mathbf{j}_{\omega,i}M^{-1}\mathbf{L}+\mathbf{j}_{\mathbf{u},i}\dot{\mathbf{u}}_i=0,\\
&\mathbf{j}_{\mathbf{v},i}=\mathbf{n}_i^T,\quad \mathbf{j}_{\omega,i}=-\mathbf{n}_i^T\left(\mathbf{p}_i-\mathbf{x}\right)^*,\quad \mathbf{j}_{\mathbf{u},i}=-\mathbf{n}_i^T.
\end{aligned}$$

Here, $(\mathbf{p}_i-\mathbf{x})^*$ represents a cross product in matrix-vector product form. Currently, we employ frictionless equality constraints, which are known to produce sticking effects at receding contacts, but in the future we plan to model inequality constraints and friction. Assembling all constraints, and substituting the output of the collision-free update and collision response, we get the condition for the collision impulses:

$$\begin{aligned}
&J_{\mathbf{v}}\delta\mathbf{v}+J_\omega M^{-1}\delta\mathbf{L}+J_{\mathbf{u}}\delta\dot{\mathbf{u}}=\mathbf{b}_\lambda,\\
&\mathbf{b}_\lambda=-J_{\mathbf{v}}\mathbf{v}^- - J_\omega M^{-1}\mathbf{L}^- - J_{\mathbf{u}}\dot{\mathbf{u}}^-. \qquad (9)
\end{aligned}$$

5.2 Constrained Collision Response

By the method of Lagrange multipliers, we define contact forces normal to the constraints, and we can arrange the constraint equation (9) and the collision updates (3) and (6) in one large linear system:

$$\begin{pmatrix} A_{\mathbf{v}} & 0 & 0 & -\Delta t J_{\mathbf{v}}^T\\ 0 & A_{\mathbf{L}} & 0 & -\Delta t J_\omega^T\\ 0 & 0 & A_{\mathbf{u}} & -\Delta t J_{\mathbf{u}}^T\\ J_{\mathbf{v}} & J_\omega M^{-1} & J_{\mathbf{u}} & 0 \end{pmatrix}\begin{pmatrix}\delta\mathbf{v}\\ \delta\mathbf{L}\\ \delta\dot{\mathbf{u}}\\ \lambda\end{pmatrix}=\begin{pmatrix}0\\0\\0\\ \mathbf{b}_\lambda\end{pmatrix} \qquad (10)$$

Substituting (3) and (6) into (9), we obtain:

$$\begin{aligned}
&A_\lambda\lambda=\frac{1}{\Delta t}\mathbf{b}_\lambda, \qquad (11)\\
&A_\lambda=J_{\mathbf{v}}A_{\mathbf{v}}^{-1}J_{\mathbf{v}}^T+J_\omega M^{-1}A_{\mathbf{L}}^{-1}J_\omega^T+J_{\mathbf{u}}A_{\mathbf{u}}^{-1}J_{\mathbf{u}}^T.
\end{aligned}$$

And we can solve for λ by Gaussian elimination. Prior to this, we anticipate the effect of constraints on the deformable body, $A_{\mathbf{u}}^{-1}J_{\mathbf{u}}^T=Y_{\mathbf{u}}$, by solving each column of $Y_{\mathbf{u}}$ separately. Knowing λ, we compute the post-collision state for both the deformable body and the visual tool.

5.3 Linear Contact Model

In the haptic thread, we can evaluate at a high update rate the discrete-time force $\tilde{\mathbf{F}}$ and torque $\tilde{\mathbf{T}}$, which comprise external forces on the tool, the dynamics of the tool itself, and the motion of the user through virtual coupling. Given $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{T}}$, plus a set of contact constraints updated at a lower rate on the visual thread, we can compute *at a high update rate* the contact force and torque that satisfy the constraints, by using the dynamically consistent inverse of the contact Jacobian [14]. Extending the concept of contact Jacobian for articulated bodies, in our case the contact Jacobian relates the generalized velocities of the rigid tool and velocities at the constraints.

Here we show how to formulate the dynamically consistent inverse of the contact Jacobian, in other words, a linear contact model that relates contact forces to discrete-time forces $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{T}}$. Recall that the method of Lagrange multipliers determines contact force $\mathbf{F}_c=J_{\mathbf{v}}^T\lambda$ and torque $\mathbf{T}_c=J_\omega^T\lambda$. Substituting λ from (11) and $\mathbf{b}_\lambda$ from (9),

$$\begin{aligned}
&\mathbf{F}_c=\frac{-1}{\Delta t}J_{\mathbf{v}}^T A_\lambda^{-1}\left(J_{\mathbf{v}}\mathbf{v}^-+J_\omega M^{-1}\mathbf{L}^-+J_{\mathbf{u}}\dot{\mathbf{u}}^-\right),\\
&\mathbf{T}_c=\frac{-1}{\Delta t}J_\omega^T A_\lambda^{-1}\left(J_{\mathbf{v}}\mathbf{v}^-+J_\omega M^{-1}\mathbf{L}^-+J_{\mathbf{u}}\dot{\mathbf{u}}^-\right). \qquad (12)
\end{aligned}$$

Substituting the collision-free update (3) in terms of $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{T}}$, we obtain the expression for our linear contact model, which will allow us to estimate contact forces from external forces in the haptic thread:

$$\begin{aligned}
&\mathbf{F}_c=\mathbf{F}_{c,0}+\frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{F}}}\tilde{\mathbf{F}}+\frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{T}}}\tilde{\mathbf{T}}, \qquad (13)\\
&\mathbf{T}_c=\mathbf{T}_{c,0}+\frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{F}}}\tilde{\mathbf{F}}+\frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{T}}}\tilde{\mathbf{T}},\\
&\mathbf{F}_{c,0}=\frac{-1}{\Delta t}J_{\mathbf{v}}^T A_\lambda^{-1}J_{\mathbf{u}}\dot{\mathbf{u}}^-,\ \mathbf{T}_{c,0}=\frac{-1}{\Delta t}J_\omega^T A_\lambda^{-1}J_{\mathbf{u}}\dot{\mathbf{u}}^-,\\
&\frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{F}}}=-J_{\mathbf{v}}^T A_\lambda^{-1}J_{\mathbf{v}}A_{\mathbf{v}}^{-1},\ \frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{T}}}=-J_{\mathbf{v}}^T A_\lambda^{-1}J_\omega M^{-1}A_{\mathbf{L}}^{-1},\\
&\frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{F}}}=-J_\omega^T A_\lambda^{-1}J_{\mathbf{v}}A_{\mathbf{v}}^{-1},\ \frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{T}}}=-J_\omega^T A_\lambda^{-1}J_\omega M^{-1}A_{\mathbf{L}}^{-1}.
\end{aligned}$$

The matrices $\frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{F}}}$, $\frac{\partial\mathbf{F}_c}{\partial\tilde{\mathbf{T}}}$, $\frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{F}}}$, and $\frac{\partial\mathbf{T}_c}{\partial\tilde{\mathbf{T}}}$ conform, precisely, the dynamically-consistent inverse of the contact Jacobian in our case. The force offsets $\mathbf{F}_{c,0}$ and $\mathbf{T}_{c,0}$ account for moving constraints. The linear contact model can be regarded as a first order approximation of contact forces in terms of external forces, where the constraints and the mass matrix do not change between time steps. Obtaining the linear model requires solving six linear systems in $A_\lambda^{-1}J_{\mathbf{v}}=Y_{\mathbf{v}}$ and $A_\lambda^{-1}J_\omega=Y_\omega$ by Gaussian elimination. On the other hand, the evaluation of contact forces in the haptic thread requires only four matrix-vector multiplications and four vector additions.

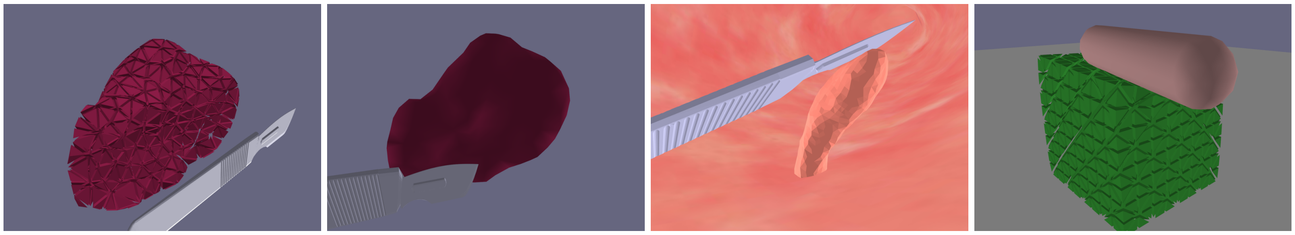

Figure 2. Test Scenarios. *From left to right:* Liver model showing the tetrahedral decomposition; Liver model being deformed; Large deformation of a polyp in a hysteroscopy simulation; Deformable block with 2560 tetrahedra and up to 25 contacts used for benchmarking.

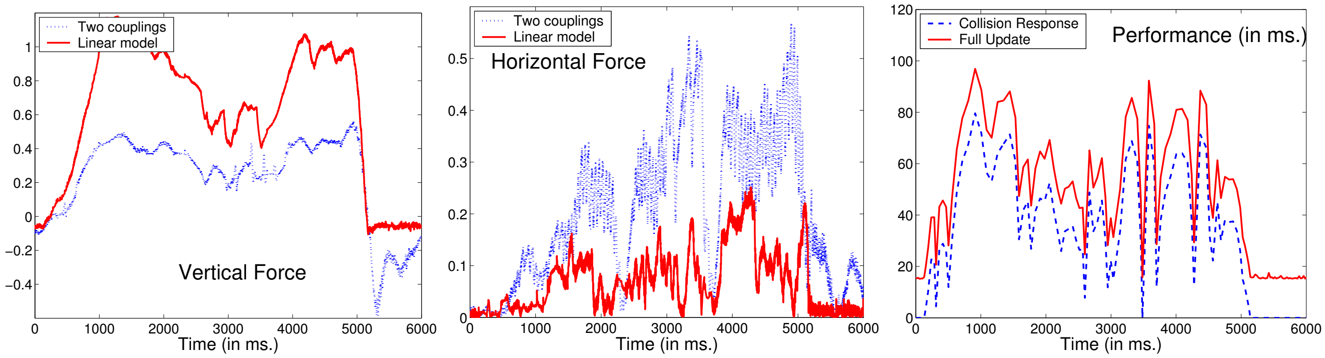

Figure 3. Forces and Timings on the Deformable Block. *Left and center:* Normal and tangential forces rendered to the user, showing enhanced transparency with our linear model, compared to a standard setup. *Right:* Cost per simulation frame of the visual thread, dominated by the computation of constraint-based collision response.

6 Results

We have tested our algorithms on a dual Pentium-4 3.0 GHz processor PC with 2.0 GB of memory, and using a PHANTOM Omni^{TM} haptic device from SensAble Technologies.

We have evaluated several approaches for solving the linear systems of equations involved in the constraint-based simulation. The large linear system (10) can be solved using a sparse iterative solver for indefinite matrices such as GMRES. Moreover, it is easy to reformulate the system in a symmetric manner, and then MINRES is also applicable. We found, however, that the convergence is worse than anticipating the constraints as in (11). Then, we are left with multiple linear systems involving the sparse SPD matrix $A_{\mathbf{u}}$. Cholesky factorization of $A_{\mathbf{u}}$ followed by multiple solutions through back-substitution seemed to be a valid option, and we tested its performance using the TAUCS library. However, we found that a conjugate gradient solver provided a faster solution in all the experiments. All the reported timings were thus obtained using conjugate gradient for the solution of sparse SPD systems.

We have applied our rendering algorithm to soft-tissue simulations for medical applications, such as contact with a deformable liver, and exploration of polyps in virtual hysteroscopy (See Figure 2). For testing the performance and transparency of the algorithm, we have designed a benchmark consisting of a cylindrical tool with 160 triangles interacting with a deformable block with 2560 tetrahedra (See Figure 2). This benchmark is simpler from the geometric point of view, but rather challenging in terms of the number of degrees of freedom of the deformable body and the number of contact points. As shown in the rightmost plot of Figure 3, the visual thread may run as slow as 10 fps, with up to 80 ms for computing constraint-based collision response. Note that computing the linear contact model is far from being the bottleneck. In this scenario, the number of contacts is as high as 25, but the multiple Gaussian eliminations needed for computing the linear contact model take a maximum of 400 μs. From this, we can conclude that using our linear contact model has a marginal cost compared to standard constraint-based simulation. The evaluation of the model in the haptic thread has a negligible cost, and the full haptic thread takes 250 μs in average per frame.

Figure 3 also shows the vertical and horizontal forces output to the user when pressing the deformable block on its top face. While we cannot exactly convey the transparency of our rendering algorithm, we can compare it to a setup with an additional virtual coupling between the haptic tool and the visual tool. Notice how the double-coupling setup suffers from large force oscillations in the horizontal force when the tool moves laterally. However, this problem does not arise with our linear contact model, which remains stable even at visual updates of only 10 fps. With such low frame rates, it is convenient to filter the parameters of the contact model that are passed to the haptic thread.

7 Future Work

We have presented a constraint-based solution for haptic rendering of contact between a rigid tool and a deformable environment. As shown by our results, a linear contact model based on the dynamically consistent inverse of the contact Jacobian enables very fast update of contact forces in the haptic thread, which together with the use of implicit integration greatly enhances rendering transparency.

We plan to extend our work by supporting inequality contact constraints and friction, possibly by solving a full LCP in the visual thread, and using the active set of constraints in the haptic thread in a way similar to Duriez et al. [9]. We also want to explore methods for accelerating the computations when the number of contacts is large. It would be interesting as well to study the loss of transparency induced by the dynamic discrepancy between the haptic-thread and visual-thread models, both analytically, as done by others [4], and from a perceptual point of view. Our preliminary evaluations indicate that there is no perceptible discrepancy when following object contours, but slow updates of the linear contact model may confuse the perception of time-of-impact during fast tool motion, and may even compromise the perceived passivity.

Our algorithm is being integrated in an already developed hysteroscopy simulator, and this will provide evaluation data in more complete settings. Similarly, we wish to test our algorithms on higher-performance haptic devices.

Acknowledgements

This work was supported by the NCCR Co-Me of the Swiss NSF. We would also like to thank the anonymous reviewers, Tanja Käser, Denis Steinemann, and Peter Leskovsky.

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

[3] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Prof. of Eurographics*, pages 57–66, 1996.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] S. Cotin, H. Delingette, and N. Ayache. Realtime elastic deformation of soft tissues for surgery simulation. *IEEE Trans. on Visualization and Computer Graphics*, 5(1), 1999.

[9] C. Duriez, C. Andriot, and A. Kheddar. A multi-threaded approach for deformable/rigid contacts with haptic feedback. *Proc. of Haptics Symposium*, 2004.

[10] 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.

[11] K. Erleben. *Stable, Robust, and Versatile Multibody Dynamics Animation*. PhD thesis, University of Copenhagen, 2004.

[12] D. L. James and D. K. Pai. Artdefo: Accurate real time deformable objects. *Proc. of ACM SIGGRAPH*, 1999.

[13] D. E. Johnson, P. Willemsen, and E. Cohen. 6-dof haptic rendering using spatialized normal cone search. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):661–670, 2005.

[14] O. Khatib. Inertial properties in robotic manipulation: An object-level framework. *Intl. Journal of Robotics Research*, 14(1), 1995.

[15] M. Mahvash and V. Hayward. High-fidelity passive force-reflecting virtual environments. *IEEE Transactions on Robotics*, 21(1):38–46, 2005.

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

[17] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. *Proc. of ACM SIGGRAPH Symposium on Computer Animation*, 2002.

[18] M. Ortega, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies. *Proc. of IEEE Virtual Reality Conference*, 2006.

[19] M. A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. *IEEE Transactions on Robotics*, 22(4):751–762, 2006.

[20] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. *Proc. of Eurographics (Computer Graphics Forum)*, 2002.