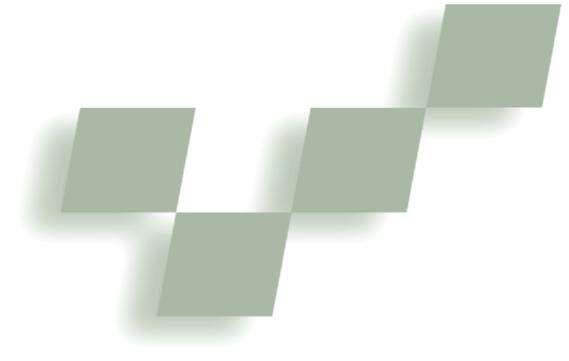


Physically Accurate Haptic Rendering with Dynamic Effects



Qi Luo and Jing Xiao
University of North Carolina, Charlotte

We present a real-time method to model and render contact force and moment between two rigid bodies, taking into account friction, gravity, and dynamic effects.

Haptic rendering simulates and renders the force or wrench (that is, force and moment) that the human operator should feel when manipulating objects in a virtual environment via a haptic device in real time. In particular, when the virtual object or tool that the operator holds (via a haptic device) contacts another virtual object, there should be a contact wrench \mathbf{W}_c exerted to the held object by the other virtual object, which affects the wrench \mathbf{W}_d exerted to the haptic device by the held object. Eventually, the haptic device exerts the wrench \mathbf{W}_h to the hand, as Figure 1 shows.

The relationship between \mathbf{W}_h and \mathbf{W}_d is device dependent—that is, the feeling of \mathbf{W}_h could be different for the same \mathbf{W}_d with different haptic devices. Simulating \mathbf{W}_c and/or \mathbf{W}_d in real time is the main task of haptic rendering. We measure the resulting performance by the rendering's realism/transparency and stability. Ideal realism/transparency requires physically accurate simulation of \mathbf{W}_c in real time and that the virtual proxy (or held object) and haptic device form a single rigid body, that is, $\mathbf{W}_c = \mathbf{W}_d$. (Note, we only consider rigid virtual objects in this article.) However, to guarantee stability, it's common to add compliance between the haptic device and the virtual proxy in terms of a virtual coupling^{1,2} at the expense of transparency, where $\mathbf{W}_c \neq \mathbf{W}_d$. We can achieve an optimal tradeoff between stability and realism or trans-

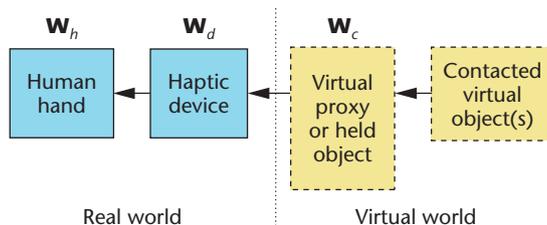
parency when we maximize virtual coupling stiffness while preserving stability.¹

This article focuses on the realism/transparency aspect of haptic rendering. We introduce a novel approach that enables physically correct and accurate simulation of contact wrench \mathbf{W}_c for general rigid objects in real time, taking into account not only friction and gravity but also dynamic effects. Our method for contact force and moment simulation builds on the real-time identification of geometrically valid contact states despite digital errors. Our approach applies to general rigid bodies including both polyhedral and nonpolyhedral objects. For nonpolyhedral, curved objects, we build our contact state representation and contact force/moment model directly on the smooth and accurate representation of the object surfaces. We use implicit or parametric equations rather than polygonal mesh approximations of the surfaces to avoid the inaccuracy caused by the artifacts of such approximation in force/moment modeling. Nevertheless, this does not prevent our method from taking advantage of the polygonal mesh representation of object surfaces for efficient collision detection³ and graphic rendering in real time.

Our approach's key idea is to solve for the contact force and moment analytically based on not only the contact configuration, but also the real-time identification of the exact type of the corresponding contact state, the type of instantaneous motion of the held object prior to reaching the contact configuration (which we call the prior motion), as well as the measured dynamic parameters associated with the prior motion. By taking into account all such information, the obtained contact force and moment are more faithful to the physical reality. By efficiently acquiring such information, we can achieve real-time computation.

To test the realism of simulation with maximum transparency, we render the effects with a SensAble Technologies Phantom 6-degrees-of-freedom device without virtual coupling, which doesn't affect stability much because our system's causality structure is an impedance display (by the Phantom device) and impedance environment (by our simulation approach).¹ In our test-

1 Effects of contact force and moment in haptic rendering.



ing experiments, rendering is stable all the time for almost all the examples.

Review of research

The real-time requirement is essential for a haptic simulation algorithm to achieve good realism, and it becomes much more stringent because the force/moment simulation must be on top of collision detection and contact identification, which are not trivial problems. To achieve real-time results, most current work on haptic rendering is either focused on single-point contact,⁴ single-area contact,⁵ or based on approximation models of objects in contact.⁶ Dynamic effects and friction are often not fully considered. These simplifications compromise physical correctness and accuracy in haptic rendering. For example, suppose that a user is virtually holding the ball in Figure 2 via a haptic device. When the held ball hits the horizontal bar—whether or not the ball spins, with or without acceleration—it should create different haptic sensations together with friction, but such differences are lost if we don't capture the ball's dynamic effects.

On the other hand, certain haptic interaction applications—such as virtual assembly, virtual prototyping, and teleoperation—require that the simulated haptic force and moment be physically correct and accurate. This in turn requires high fidelity in modeling contact forces and moments.

Several researchers have worked on physically accurate contact force and moment computation using analytical methods in dynamic simulation.⁷⁻⁹ These methods build up constraint equations that require contacting objects to never interpenetrate. The equations then must be solved numerically for the contact forces and moments, which guarantee that those constraint equations are satisfied. However, solving constrained equations as a linear complementarity problem is time consuming. For cases involving complex contact states and objects, real-time computation often isn't possible. Thus, such methods are inadequate for haptic rendering.

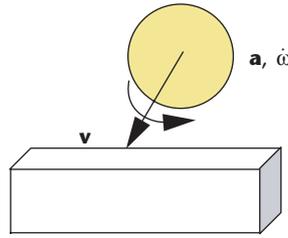
Moreover, the contact simulation problem in dynamic simulation is to solve the velocities of objects after contact given the force/moment and the velocities of the objects before contact. A collided virtual object acts as an admittance to object A colliding with it—accepting forces/moments from A and returning velocities of A—rather than as an impedance. The respective methods (even if some can achieve real-time computing in certain cases²) are not applicable to haptic rendering with an impedance display and impedance virtual environment, where the force/moment on the held virtual object A prior to a contact isn't entirely known before the contact happens.

Assumptions and general notations

Here we introduce basic assumptions and notations used in our work.

Physical properties of objects

We consider rigid objects with evenly distributed mass. We also assume evenly distributed stiffness with a constant stiffness coefficient K and evenly distributed con-



2 Dynamic effects and haptic sensation. Realistic modeling of the haptic sensation should take into account the accelerations \mathbf{a} and $\dot{\omega}$ of the ball.

tact pressure. We use Coulomb friction with the static friction coefficient μ and kinetic friction coefficient μ_D .

Geometric properties

Principal contacts (PCs) describe contact primitives between two polyhedral objects in terms of the surface elements in contact.¹⁰ A surface element can be a face, an edge, or a vertex. A face's boundary elements are the edges and vertices bounding it, and an edge's boundary elements are the vertices bounding it. Formally, a PC denotes the contact between a pair of surface elements that are not boundary elements of other contacting surface elements. This ensures that PCs are the highest level contact primitives to describe a contact state. For example, a face-face contact between two polyhedral objects is described just as a single face-face PC rather than in terms of a set of vertex-face or edge-face contacts. Each PC defines a single contact region of a point, a straight-line segment, or a plane segment.

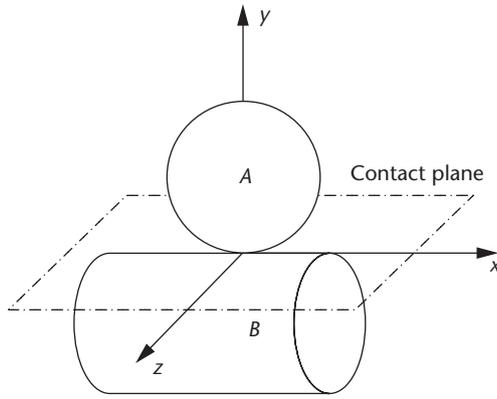
For nonpolyhedral objects, a face is a smooth surface or surface patch defined by a parametric function $p(u, v)$, where $p = [xyz]^T$, and/or implicit surface equation $f(x, y, z) = 0$. An edge is a smooth intersection curve of two faces, which can be described either parametrically or as satisfying both implicit surface equations of the two faces. A vertex is either an apex of a surface (such as a cone) or an intersection point of two or more edges, which satisfies both implicit edge equations of those edges. Both faces and edges can be closed or finite with boundary elements. Note that points of discontinuities on a face are on its bounding edges and vertices, and points of discontinuities on an edge are its bounding vertices.

We restrict our consideration of contacts involving curved surface elements of rigid objects to cases where a contact region consists of either a point (which is common between two faces of different curvatures), a straight-line segment (which can occur between two revolute objects, such as between two cylindrical faces), or a planar area (which can occur between a curved line and a flat surface or between two flat surfaces). Now we can extend the definition of a PC so that it describes such a contact region in terms of the two contacting surface elements and the tangent plane of the contact region, which we call the *contact plane*. There are three categories of PCs based on the types of contact regions:

- *Point PC*: the contact region is a point.
- *Line PC*: the contact region is a straight line segment.
- *Plane PC*: the contact region is a planar area.

Built on this extended notion of PCs, we can describe a

3 Example task frame with respect to a principal contact.



general contact state as a *contact formation (CF)* consisting of the set of PCs formed.

Objects and motions

Consider A as the held (moveable) object and B as the fixed object(s) in a virtual environment. When A hits B during its motion, only one PC can be gained at a time, because the probability of gaining two PCs simultaneously is extremely low in reality. Furthermore, at the moment a PC is gained, the corresponding contact region of A either sticks or slides but does not rotate about an axis along the normal of the contact plane. More than one PC can be broken at the same time. Once in contact, A can also perform any kind of compliant motion with respect to B.

Object and task frames

Let *A* also denote the object frame of A with its origin set at the center of mass of A, and *B* denote the object frame of B. Then the configuration of A at any instant is described by the homogenous transformation matrix ${}^B\mathbf{T}_A$ from A to B.

When A and B are in a contact formation $CF = \{PC_i | i = 1, \dots, n\}$, we establish a PC-based task frame according to each PC in the CF as the following: The origin of the frame is a contact point on the contact plane of the PC, the y-axis is along the normal of the contact plane pointing to the held object A, and the x- and z-axes are along the tangent of the contact plane and orthogonal to each other, following the right-hand rule (see Figure 3).

Contact force and moment representation

Given A and B in a $CF = \{PC_i | i = 1, \dots, n\}$, we use the PC_i -based task frame to describe the contact force to A from B at PC_i , denoted as ${}^i\mathbf{F}_c$, which can be decomposed into a normal force along the +y-axis, ${}^i\mathbf{F}_{cy}$, and a friction force along the xz-plane (the contact plane) of the *i*th task frame, ${}^i\mathbf{F}_{cf}$.

For a point PC_i , ${}^i\mathbf{F}_c$ applies to the contact point p_{i0} on A. We can express the contact moment with respect to A's origin (center of mass) as ${}^A\mathbf{M}_i = {}^A\mathbf{p}_{i0} \times {}^A\mathbf{R}_i {}^i\mathbf{F}_c$, where ${}^A\mathbf{p}_{i0}$ is the position vector of p_{i0} in frame A, and ${}^A\mathbf{R}_i$ denotes the rotation transformation matrix from the PC_i task frame to frame A.

For a line or plane PC_i , we can consider the contact force as applied to either one equivalent contact point p_{i0}

or more equivalent contact points $p_{ij}, j = 1, 2, \dots$, on A (see the “Equivalent Points of Contact and Contact Force Distribution” sidebar). If only one equivalent contact point exists, we compute the contact moment just like we would in the case of a point PC. In the case of more equivalent contact points, the contact forces are distributed to each equivalent point as ${}^i\mathbf{F}_{cj}$ and ${}^i\mathbf{F}_c = \sum_j {}^i\mathbf{F}_{cj}$. The contact moment is thus ${}^A\mathbf{M}_i = \sum_j {}^A\mathbf{p}_{ij} \times {}^A\mathbf{R}_i {}^i\mathbf{F}_{cj}$.

We can now express the total contact force \mathbf{F}_c to A by B as $\mathbf{F}_c = \sum_i {}^A\mathbf{R}_i {}^i\mathbf{F}_c$, and the total contact moment \mathbf{M}_c to A by B can be expressed as $\mathbf{M}_c = \sum_i {}^A\mathbf{M}_i$.

Contact force modeling

At any instant when the held A contacts B, the total contact force exerted to A by B depends on not only A's configuration but also the current CF between A and B and the prior motion of A before reaching the current contact configuration with its dynamic effects.

Real-time identification of CFs

Real-time identification of CFs (especially those consisting of more than one PC) in the virtual world is not trivial because of the ambiguity introduced by digital computation errors. Figure 4 shows an example, where there are several possibilities of PCs for each contact region. If we consider just PC_1 (or PC_2) in isolation, then it does not matter which type of PC is identified because either one is a geometrically valid PC. However, here we have to consider both PC_1 and PC_2 together. Thus, if PC_1 is the edge–face PC, PC_2 has to be the vertex–face PC, for example, to not violate the geometric contact constraint. The combination of a face–face PC_1 and a vertex–face PC_2 , on the other hand, is not a valid CF.

Our approach to identify a valid CF involves

- identifying all possible PCs based on the result of real-time collision detection and
- finding a combination of PCs that form a geometrically valid CF by looking up a table of valid CFs built offline beforehand.

For polyhedral objects, we can build such a table automatically.¹⁰ The number of CFs is polynomial to the maximum number of edges and vertices bounding a face at most and is in the order of hundreds for many practical cases of polyhedral objects.¹⁰ For comparable objects with curved surfaces, usually far fewer valid CFs exist, and thus table building can be much simpler. For example, in the case of Figure 5, there are only three valid multi-PC CFs between A (a sphere) and B (two cylinders), unlike the comparable case of polyhedral objects in Figure 4. In our experiments (see the “Implementation” section), the time for searching valid CFs was almost negligible compared to the time for collision detection, and we were able to achieve real-time identification of CFs.

Prior motion types

Without losing generality, let t^- and t^+ indicate the two consecutive time instants sampled immediately before and after A reaches a contact configuration X. We call

Equivalent Points of Contact and Contact Force Distribution

Consider that A has a PC, PC_i , with object B in CF^+ at t^+ . If PC_i is a line PC, let E denote the contact line segment on A. Since the motion of E from t^- to t^+ is very small (almost instantaneous), we can approximate it as either having no rotation about an axis r_i normal to the contact plane of PC_i — E either translates or is static along the contact plane (case 1)—or a pure rotation with r_i through an internal point o_{ri} of E (case 2).

In case 1, we can obtain the total effect of contact force and moment on E by concentrating all forces on a single equivalent point of contact p_{i0} at the center of E since all contact friction forces on E are in the same direction (all contact normal forces on E are always in the same direction).

In case 2, the point o_{ri} divides E into two subsegments, where the friction forces on each subsegment have the same direction, which is opposite to the direction of the friction forces of the other subsegment. We can thus obtain the total effect of contact force and moment on E by concentrating the contact forces at a pair of equivalent points of contact p_{i1} and p_{i2} , at the centers of the subsegments, respectively. We can further achieve this by first distributing the magnitude of the contact normal force ${}^iF_{cy}$ on p_{i1} and p_{i2} , which we can reasonably approximate as proportional to the distances l_1 and l_2 from p_{i1} and p_{i2} respectively to o_{ri} —that is, ${}^iF_{c1y} = {}^iF_{cy}l_1/(l_1 + l_2)$, ${}^iF_{c2y} = {}^iF_{cy}l_2/(l_1 + l_2)$. ${}^iF_{c1y}$ and ${}^iF_{c2y}$ are along the same direction as ${}^iF_{cy}$. Next we can compute the contact friction forces concentrated at each p_{ij} from ${}^iF_{c1y}$, $j = 1, 2$, and reasonably approximate them as kinetic frictions.

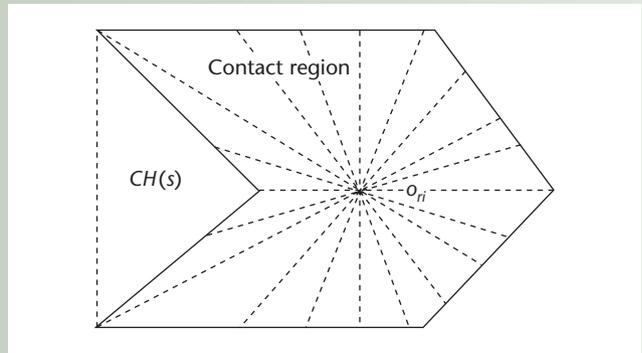
If PC_i is a plane PC, let S denote the contact region on A

and $CH(S)$ be the convex hull of S . We can also approximate the motion of S from t^- to t^+ as either having no rotation about an axis r_i normal to the contact plane of PC_i —that is, S either translates or is static along the contact plane (case 3)—or a rotation with r_i through an internal point o_{ri} of $CH(S)$ (case 4).

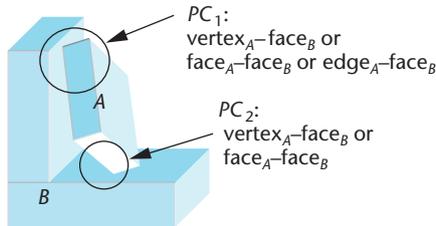
If case 3, a single equivalent point of contact p_{i0} at the centroid of $CH(S)$ is sufficient, just as in case 1 for a line PC.

In case 4, we can approximate the effect of contact force and moment on S as that of a set of m equivalent line PCs formed by evenly distributed line segments of $CH(S)$ through o_{ri} (Figure A). We can consider m as a tradeoff between accuracy and real-time computation requirements.

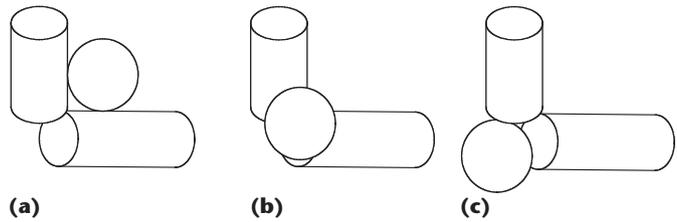
${}^iF_{cy}$ should be first distributed on the m equivalent line PCs according to the proportions of their corresponding contact line segments. Next, the portion of ${}^iF_{cy}$ distributed to each line PC is further distributed to the corresponding pair of equivalent points of contact proportional to their distances to o_{ri} .



A Equivalent line PCs to approximate a plane PC.



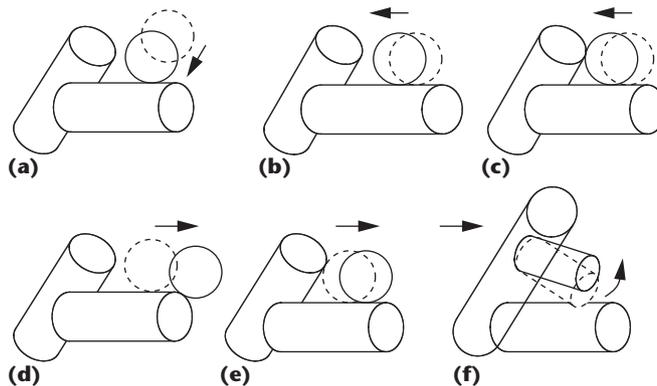
4 Ambiguity in identification: PC_1 has three possibilities and PC_2 has two possibilities.



5 Three valid 2-PC contact formations. (a) face-face, face-face; (b) face-edge, face-edge; and (c) face-face, face-face.

A's motion during time frame $\Delta t = [t^-, t^+]$ the instantaneous motion of A prior to X, or the instantaneous prior motion, which is one of the following types (as shown in Figure 6):

- MT1: guarded motion from no contact at t^- to a PC at t^+ .
- MT2: no motion or compliant motion to maintain a CF.
- MT3: compliant motion to gain a new PC.
- MT4: compliant motion to change or break PCs.



6 Examples of prior motion types (MTs). (a) MT1, (b) MT2, (c) MT3, (d) MT4 (change PC), (e) MT4 (break PC), and (f) MT4 (change a PC and break a PC).

Moment of Inertia

Given the moments of inertia about the three axes of A's coordinate system $o_A - xyz$, I_{xx} , I_{yy} , and I_{zz} , and the products of inertia, I_{xy} , I_{xz} , and I_{yz} (all of which can be computed offline), we can obtain the moment of inertia of A about an arbitrary axis through o_A with unit vector $\mathbf{n} = [\cos \alpha, \cos \beta, \cos \gamma]^T$ as follows:

$$I_{nn} = I_{xx} \cos^2 \alpha + I_{yy} \cos^2 \beta + I_{zz} \cos^2 \gamma + 2I_{xy} \cos \alpha \cos \beta + 2I_{yz} \cos \beta \cos \gamma + 2I_{xz} \cos \alpha \cos \gamma \tag{A}$$

Therefore, we can compute I_{nn} in real time.

For certain simple objects with symmetry, by properly establishing $o_A - xyz$, there are cases where the products of inertia are all zero, further simplifying Equation A.

We can detect these prior motion types in real time based on

- the change of object A's configuration from t^- to t^+ , which can be expressed by the homogeneous transformation matrix ${}^w\mathbf{T}_+$ consisting of a position vector ${}^w\mathbf{p}_+$ and a rotation matrix ${}^w\mathbf{R}_+$, and
- the CF at t^+ , CF^+ , and the prior state of A at t^- , which could be either no contact or also a CF, CF^- .

In addition, the dynamic parameters associated with each prior motion—that is, linear and angular velocities \mathbf{v} and ω and accelerations \mathbf{a} and $\dot{\omega}$ —can be tracked, as A moves, after the initial two time frames. Specifically,

$$\mathbf{v} = \mathbf{v}(t^-) = {}^w\mathbf{p}_+ / \Delta t, \mathbf{a} = \mathbf{a}(t^-) = [\mathbf{v}(t^+) - \mathbf{v}(t^-)] / \Delta t$$

From the rotation matrix ${}^w\mathbf{R}_+$, we can obtain the rotation axis (through A's origin) and roll-pitch-yaw angle increments, and thus obtain the angular velocity ω . We can obtain $\dot{\omega}$ subsequently from $\omega(t^-)$ and $\omega(t^+)$.

From the prior accelerations, we can compute the external force and torque exerted on A at t^- . Let \mathbf{F}_a denote the external force acted on A and computed from \mathbf{a} . Let τ denote the external torque on A computed from the angular acceleration $\|\tau\| = |I \cdot \dot{\omega}|$, where I is the moment of inertia of A with respect to the instant rotation axis. We show the derivation of I in the "Moment of Inertia" sidebar.

Such prior external force and torque affect the contact force and torque on A at t^+ associated with $CF^+ = \{PC_1, \dots, PC_n\}$.

Contact normal force

Recall that PC_i , $i = 1, \dots, n$, describes a pair of contacting surface elements, which means that the minimum distance d_i between the corresponding surface elements of PC_i is within a small threshold $\epsilon > 0$. We compute the magnitude of the contact normal force ${}^iF_{cy}$ (that is, along the +y-axis of the task frame i) at PC_i based on a spring-damper model as

$${}^iF_{cy} = K(\epsilon - d_i) - Dv_y$$

where K is the stiffness coefficient, D is the coefficient

of damping, and v_y is the y-component of the prior linear velocity of the task frame. To prevent penetration, we can also use the constraint-based idea of a virtual proxy.⁴ Depending on the type of PC_i and A's motion, we can further distribute ${}^iF_{cy}$ to equivalent contact point(s) (see the "Equivalent Points of Contact and Contact Force Distribution" sidebar).

Contact friction forces

We now describe how to determine the contact friction force applied to each contact (or equivalent contact) point p_{ij} ($j = 0$ or $j = 1, 2, \dots$) of object A at each PC_i , given ${}^w\mathbf{T}_+$ (which consists of ${}^w\mathbf{p}_+$ and ${}^w\mathbf{R}_+$), CF^+ , CF^- (if it existed), and based on the prior motion of A with corresponding dynamic parameters.

Prior motion type is MT1. In this type of prior motion, the held object A moved from no contact at t^- to forming a single PC (PC_i) at t^+ . In this case, either there is a single point of contact (if PC_i is a point PC) or a single equivalent point of contact of A since the contact region at PC_i does not rotate at t^+ .

Let's denote the point as p_{i0} and consider the most general case when the prior motion of A has both linear and angular velocities \mathbf{v} and ω and accelerations \mathbf{a} and $\dot{\omega}$. The derived results will also work for special cases when certain velocity and/or acceleration parameters are zero.

First, consider the forces acted upon the contact point p_{i0} of A. Let ${}^i\mathbf{F}_a$ be the force result of prior acceleration \mathbf{a} of A expressed in the task frame of PC_i . Then, let ${}^i\mathbf{F}_\tau$ be the force effect of τ (the torque resulted from prior angular acceleration of A, computed in the "Prior motion types" section). Next, $\|{}^i\mathbf{F}_\tau\| = \|\tau\|/r$, where r is the distance from p_{i0} to the rotation axis along the direction of τ through A's origin o_A . We can determine the direction of ${}^i\mathbf{F}_\tau$ by $\omega \times \mathbf{p}_{i0}$, where \mathbf{p}_{i0} is the position vector of p_{i0} in frame A.

Now let ${}^i\mathbf{F}_c$ denote the contact force on p_{i0} from object B, which has a normal component ${}^i\mathbf{F}_{cy}$ (as we described in the previous section) and a friction component ${}^i\mathbf{F}_{cf}$. The friction component ${}^i\mathbf{F}_{cf}$ is what we need to discern.

Next, let's consider the force that the contact point p_{i0} of A applies to B. It is the sum of ${}^i\mathbf{F}_a$, ${}^i\mathbf{F}_\tau$, which are known from the prior motion of A, and an impulse force ${}^i\mathbf{F}_v$ upon contact, which are unknown from the prior motion.

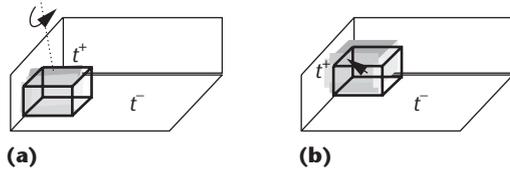
Based on Newton's third law, the force that p_{i0} applies to B is the opposite of the (contact) force that B applies to p_{i0} , and thus we have

$${}^i\mathbf{F}_c = -({}^i\mathbf{F}_a + {}^i\mathbf{F}_\tau + {}^i\mathbf{F}_v) \tag{1}$$

Now let ${}^i\mathbf{u}_0 = [a \ b \ c]^T$ be the unit vector denoting the direction of the prior linear velocity ${}^i\mathbf{v}_0$ of the contact point p_{i0} represented in the task frame of PC_i , which is the sum of $\mathbf{v} + \mathbf{v}_\omega$, where \mathbf{v}_ω is the prior linear velocity caused by ω at p_{i0} .

Assume p_{i0} is stuck upon contact. Then the direction of ${}^i\mathbf{F}_v$ is along ${}^i\mathbf{u}_0$. That is, we can rewrite Equation 1 as the following three linear equations:

$$\begin{bmatrix} {}^i F_{cx} \\ {}^i F_{cy} \\ {}^i F_{cz} \end{bmatrix} = - \begin{bmatrix} {}^i F_{ax} \\ {}^i F_{ay} \\ {}^i F_{az} \end{bmatrix} - \begin{bmatrix} {}^i F_{\tau x} \\ {}^i F_{\tau y} \\ {}^i F_{\tau z} \end{bmatrix} - \|{}^i \mathbf{F}_v\| \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2)$$



7 Examples of no actual physical motion. The indicated rotation and translation in (a) and (b) respectively from t^- to t^+ are not possible.

Only three items in Equation 2 are unknown: ${}^i F_{cx}$, ${}^i F_{cz}$, and $\|{}^i \mathbf{F}_v\|$, which we can solve from the known items in Equation 2.

If

$$({}^i F_{cx})^2 + ({}^i F_{cz})^2 \leq (\mu {}^i F_{cy})^2$$

then p_{i0} is stuck upon contact, and the friction force at p_{i0} is ${}^i \mathbf{F}_{cf} = [{}^i F_{cx} \ 0 \ {}^i F_{cz}]^T$. Otherwise, p_{i0} slides immediately upon contact along the projection of ${}^i \mathbf{u}_{0xz}$ on the contact plane (the xz -plane of frame i). Let ${}^i \mathbf{u}_{0xz}$ be the unit vector of the following projection:

$${}^i \mathbf{u}_{0xz} = \begin{bmatrix} a & 0 & c \end{bmatrix}^T / \sqrt{a^2 + c^2}$$

Then the friction force at p_{i0} is

$${}^i \mathbf{F}_{cf} = -\mu_D {}^i F_{cy} {}^i \mathbf{u}_{0xz}$$

Prior motion type is MT2. In this case, the CF is maintained—that is, $CF^+ = CF^- = CF = \{PC_i, i = 1, \dots, n\}$. Knowing that, we need to further identify whether from t^- to t^+ ,

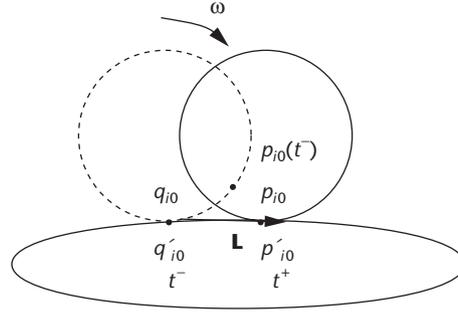
- *Case a:* A has no actual physical motion.
- *Case b:* A's motion involves pure rolling.
- *Case c:* A's motion is a pure rotation about certain fixed contact point(s) on A.
- *Case d:* A's motion makes all contact points slide.

We identify cases a through d in turn by elimination. That is, only if case a is not true, we consider case b, and only if case b is not true, we consider case c, and so on.

There are two situations when case a is true. First, if there is hardly any change of A's configuration from t^- to t^+ —that is, ${}^i \mathbf{T}_+$ is approximately an identity matrix—then case a is true. Second, if ${}^i \mathbf{T}_+$ does not correspond to a possible physical motion satisfying the contact constraints, then case a is true. This is the case when the equivalent axis of rotation $\hat{\mathbf{u}}$ computed from ${}^i \mathbf{R}_+$ is not (approximately) along a possible axis of rotation to maintain CF (see Figure 7a) or else ${}^i \mathbf{p}_+$ is not (approximately) along a possible translation axis to maintain CF (see Figure 7b).

If case a is true, we consider all contact points at every PC_i static. Either there is one contact point (if PC_i is a point PC) or there is a single equivalent point of contact. Let's denote that point as p_{i0} . Let \mathbf{s}_{pi0} denote the displacement vector of p_{i0} from t^- to t^+ along the contact plane of PC_i . Note that when case a is true, $\|\mathbf{s}_{pi0}\|$ is smaller than a very small threshold. To compute the corresponding static friction, we can do the following. If

$$K_p \|\mathbf{s}_{pi0}\| \leq \mu {}^i F_{cy}$$



8 Illustration of rolling: the distance L from q'_{i0} to p'_{i0} along the surface of B equals the distance between q_{i0} and p_{i0} along the surface of A.

we can compute the static friction as

$${}^i \mathbf{F}_{cf} = -K_p \mathbf{s}_{pi0} \quad (3)$$

where K_p is the proportional control gain.⁴ Otherwise, ${}^i \mathbf{F}_{cf}$ is set to be the maximum static friction:

$${}^i \mathbf{F}_{cf} = -\mu {}^i F_{cy} \mathbf{s}_{pi0} / \|\mathbf{s}_{pi0}\| \quad (4)$$

If not case a, then we check if A's motion is of case b. Case b is possible only if the following conditions hold: either A or B is a curved object; CF has at most two PCs, and at least one of them is a nonvertex point PC or a nonedge line PC, which are the only types of PC where pure rolling may happen.

Consider a nonvertex point PC_i of CF , with the contact point p_{i0} of A. If p_{i0} is in contact with p'_{i0} of B, and it is not the contact point q_{i0} of A at time t^- , which contacts q'_{i0} of B, and if the distance L from q'_{i0} to p'_{i0} along the surface of B (see Figure 8) approximately equals the distance between q_{i0} and p_{i0} along the surface of A, then we say, rolling happens, and the friction force ${}^i \mathbf{F}_{cf}$ at p_{i0} is static. We can determine the following: If $L \leq \mu {}^i F_{cy} / K_p$, then $\|{}^i \mathbf{F}_{cf}\| = K_p L$; else $\|{}^i \mathbf{F}_{cf}\| = \mu {}^i F_{cy}$ (the maximum static friction) with the force direction along \mathbf{s}_{pi0} .

Similarly, for a nonedge line PC_i , we can pick the centroid of the contact line segment of A, call it p_{i0} , and use the same checking as noted previously to decide if rolling happens. If so, since the axis of rotation does not go through any internal points of the contact line segment, p_{i0} is the single equivalent point of contact at PC_i . We can compute the static friction applied at p_{i0} in the same way as we did for rolling at a point PC.

Determining the Planar Rotation's Axis

Consider a planar rigid object G (which could be a line segment) on a plane P , and p is a point on G . The general planar motion of G during a small time interval $\Delta t = [t^-, t^+]$ is equivalent to G rotating about a certain axis \mathbf{r}_r normal to P through a point o_r on P . Denote the angular velocity of G at t^- as ω and the distance from $p(t^-)$ to $p(t^+)$ as δ . We can determine the position of o_r as having equal distance d to both $p(t^-)$ and $p(t^+)$ so that $d = \delta / (\omega \Delta t)$. If the rotation from $p(t^-)$ and $p(t^+)$ is counterclockwise, o_r is on the left-hand side of the curve from $p(t^-)$ and $p(t^+)$. Otherwise, o_r is on the right-hand side of the curve. If o_r is on G , G has a pure rotation about o_r . Otherwise, if o_r is outside of the convex hull of G , then G approximately has a pure translation during Δt .

If CF consists of two PCs and rolling happens at one PC, then the contact friction effect at the other PC is kinetic. Depending on the type of PC, we can determine one or more contact or equivalent contact points. Without losing generality, let $p_{ij}, j = 0$ or $1, 2, \dots$, and let ${}^iF_{c_{ij}}$ be the amount of contact normal force distributed to p_{ij} . Let $\mathbf{s}_{p_{ij}}$ denote the displacement vector of p_{ij} from t^- to t^+ along the contact plane of PC_i . The contact friction ${}^jF_{c_{ij}}$ at p_{ij} is kinetic and satisfies

$${}^iF_{c_{ij}} = -\mu {}^iF_{c_{ij}} \mathbf{s}_{p_{ij}} / \|\mathbf{s}_{p_{ij}}\| \quad (5)$$

If not case b, we consider if case c holds. From \mathbf{T}_+ , we can obtain the direction \mathbf{u}_r of the rotation axis \mathbf{r} . The necessary conditions that a pure rotation with an axis through certain contact point(s) can maintain CF include

1. CF consists of at most two PCs.
2. If CF consists of two point PCs, \mathbf{u}_r is parallel to the line connecting the two contact points.
3. If CF contains two PCs and at least one of them is not a point PC, \mathbf{u}_r is perpendicular to the contact plane of at least one PC.
4. If CF contains a line PC, \mathbf{u}_r is either perpendicular or parallel to the contact line segment of the PC.

Case c is true if these four conditions hold, and we can find the actual rotation axis \mathbf{r} through certain contact point(s).

We now describe how to decide whether \mathbf{r} is through certain contact point(s) for each PC_i in CF , given that conditions 1 through 4 are satisfied, and how to compute the contact friction forces at each PC_i accordingly.

If PC_i is a point PC, and the contact point p_{i0} hardly moved from t^- to t^+ , we say p_{i0} is static and \mathbf{r} is through it. We can compute the friction force at the static p_{i0} as in case a, which satisfies either Equation 3 or 4. Otherwise, \mathbf{r} is not through p_{i0} , and the friction force at p_{i0} is kinetic and satisfies Equation 5.

If PC_i is a line PC, \mathbf{u}_r is parallel to the contact line segment, and the line segment almost did not translate from t^- to t^+ , then we say the contact line segment is static and \mathbf{r} is on it. The contact force and moment effect at PC_i can be equivalent to that of a single equivalent point of contact, so we can also call it p_{i0} . We can compute the

friction force at the static p_{i0} as in case a. If \mathbf{r} is not through p_{i0} , the friction force at p_{i0} is kinetic and satisfies Equation 5.

If PC_i is a line PC, \mathbf{u}_r is perpendicular to its contact plane, and \mathbf{r} goes through a point inside the contact line segment (the “Determining the Planar Rotation's Axis” sidebar shows how to detect that), the contact force and moment effect at PC_i can be obtained as that applied to a pair of equivalent points of contact, $p_{ij}, j = 1, 2$. The contact friction ${}^jF_{c_{ij}}$ at p_{ij} is kinetic and satisfies Equation 5. If \mathbf{r} goes through a point outside the contact line segment, then a single equivalent point of contact p_{i0} is sufficient to express the effect of contact forces and moments, and the friction force at p_{i0} is kinetic and satisfies Equation 5.

If PC_i is a plane PC (\mathbf{u}_r is perpendicular to its contact plane), then \mathbf{r} either goes through a point inside the contact region, or inside the convex hull of the contact region, or outside the convex hull (see the “Determining the Planar Rotation's Axis” sidebar). We can obtain the contact force and moment effect at PC_i as that of multiple equivalent points of contact, $p_{ij}, j = 1, 2, \dots$, and at each p_{ij} , the contact friction is kinetic and satisfies Equation 5. If \mathbf{r} is through at least one contact point of at least one PC, case c is true.

If case c does not hold, the only case left is case d, where the friction effect at every PC is kinetic. Depending on the type of PC, one or more contact or equivalent contact points can be determined, and at each (equivalent) contact point, we can compute the friction force by Equation 5.

Prior motion type is MT3. If PC_k is the newly gained PC in CF^+ , we can compute the friction force at the (equivalent) contact point of PC_k just as we described in MT1. If A becomes static at PC_k , the friction effect at each $PC_i, i \neq k$, at t^+ , should be static as in case a of MT2. Otherwise, the friction effect at each PC_i should be dynamic as in case d of MT2.

Prior motion type is MT4. If from CF^- to CF^+ the transition cannot be made by pure rolling or a compliant rotation about fixed contact point(s), the friction effect at each PC in CF^+ is dynamic as in case d of MT2. Otherwise, we can determine the friction effect as in case b or c of MT2.

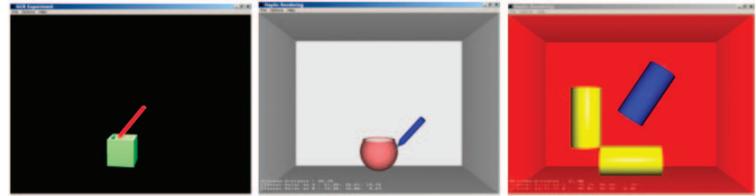
Haptic force and moment

When contact happened, with the total contact force \mathbf{F}_c determined, the haptic force \mathbf{F}_d that the haptic device (and subsequently the human operator) should feel can be readily obtained as the sum of the contact force and the gravity force: $\mathbf{F}_d = \mathbf{F}_c + \mathbf{F}_g$, and can be rendered accordingly. The haptic moment \mathbf{M}_d felt by the device (and the human operator) at any instant equals the combined moment created by the contact force upon A and the gravity of A . If \mathbf{M}_d is with respect to A 's origin, which is also A 's center of mass, then, $\mathbf{M}_d = \mathbf{M}_c$.

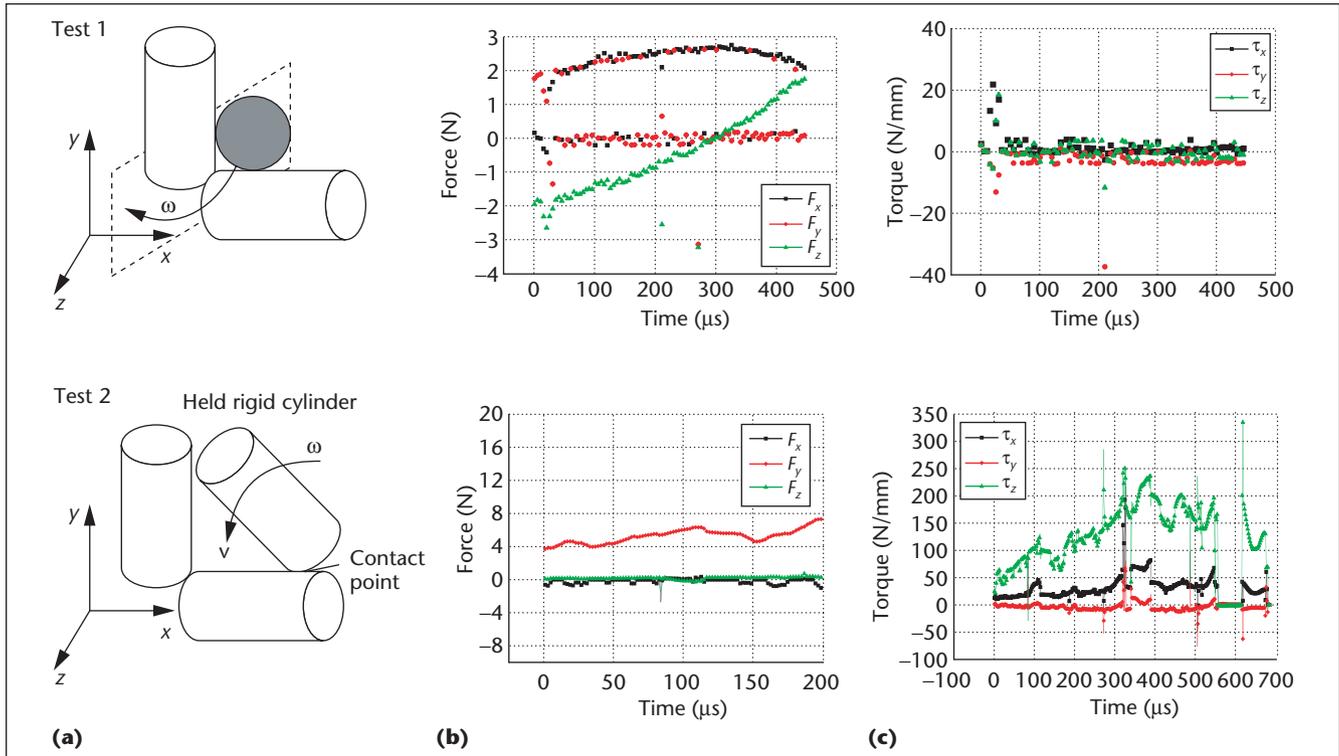
When the held object is free of contact and moving with acceleration \mathbf{a} and angular acceleration $\dot{\omega}$ (which we can track), we can compute the haptic force and moment easily: $\mathbf{F}_d = \mathbf{F}_g - m\mathbf{a}$, $\mathbf{M}_d = -I\dot{\omega}$.

Table 1. Values for the parameters of our implementation.

Value	Parameter
g	9.8 N/kg
μ	0.7 N/kg
μ_D	0.08 N/kg
D	0.005 N*sec./mm
K, K_p	2.3, 1.6 N/mm
ε	1.0 mm



9 Examples of haptic rendering: (a) peg and hole, (b) pen and bowl, and (c) cylinder and pipe.



10 Compliant motions of (a) two tests, (b) contact forces, and (c) torque components.

Implementation

We implemented this method and applied it to real-time haptic rendering involving both arbitrary polyhedral objects and curved objects. We used a Phantom 6-DOF device, which was connected to a personal computer with dual Intel Xeon 2.4 GHz processors and 1 Gbyte of RAM. The human operator virtually held a rigid object A by attaching it to the haptic device and made arbitrary contact to the fixed object B. The operator also performed arbitrary compliant and guarded motions.

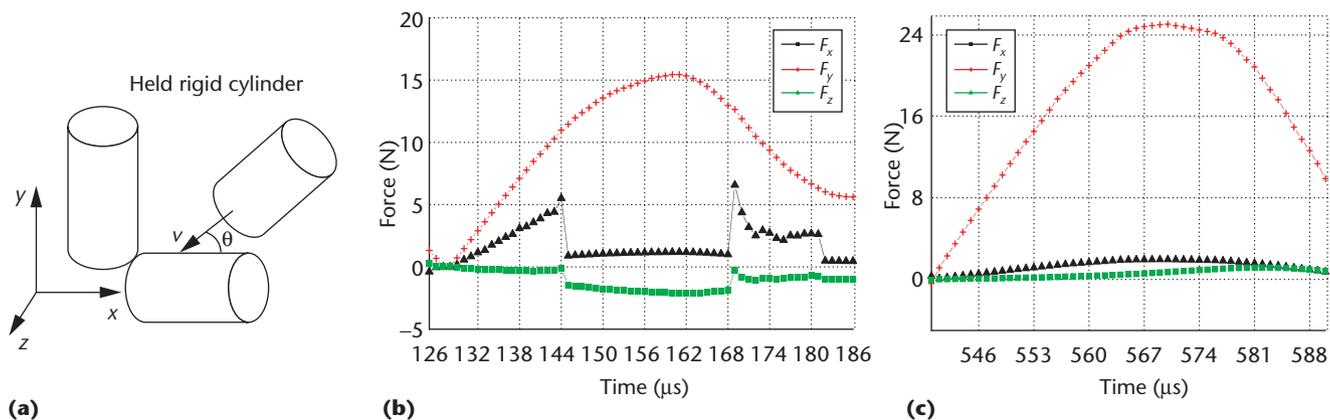
Although our method of computing haptic force/moment used smooth parametric representation of the object geometry, we used polygonal mesh representation for curved objects for fast graphic rendering. To enable smooth and stable rendering across time frames, we applied interpolation or shading^{4,6} to contact forces when needed. Table 1 shows the values we used for the parameters of our implementation.

Figure 9 shows some implemented examples with our haptic rendering method. Figure 9a shows a polyhedral

peg A and hole B, Figure 9b shows a curved pen A and bowl B (where the bowl is made of parabolic and spherical surfaces), and Figure 9c shows a cylinder A and a pipe B (formed by two more cylinders in an L-shape). For collision detection, we used the algorithm reported by Zhang and Xiao between polyhedra¹¹ in Figure 9a and the algorithm reported by Zou and Xiao between curved objects¹² in Figures 9b and 9c.

Figure 10 and Figure 11 (next page) show some test results, where the units of force and torque are Newton and N/mm, respectively. The unit of time is approximately 10 μs.

Figure 10 shows two tests. In test 1, A is the rigid ball (with mass 0.5 kg and radius 20 mm) and B is the cylinder. A slides along both cylinders compliantly (keeping contact with both). As expected, the force value F_z changes from negative to positive, while both F_x and F_y first increase and then decrease. The horizontal bar in the force chart shows zero contact force because in the actual operation, contacts are sometimes briefly broken and rebuilt again.



11 Guarded motion test to demonstrate the influence of dynamic effects.

Some scattered outlier force values also indicate such transitions.

In this test, there are recognizable torque values including some scattered large ones. This happens because when a contact is broken briefly and rebuilt again, sometimes the ball is stuck immediately upon collision and the friction force becomes static (as analyzed for the prior motion type of MT1 and MT4), which is much greater and therefore results in a much larger torque. Notice also that just as expected from the effect of the static friction, τ_x is mostly positive, τ_y is mostly negative, and τ_z has both positive and negative values. Recall that the torques are about the center of the ball.

Test 2 shows a case where A is a rigid cylinder (with mass 0.5 kg, radius 20 mm, and length 80 mm), and it rotates about a fixed point against another cylinder. The contact normal force F_y is almost constant, while F_x has recognizable negative values because the friction force is static now. Note that the torque τ_z is caused by F_y and F_x together. τ_z increases as A rotates in the direction shown because of the increase in the horizontal component of the distance from the center of the cylinder to the contact point.

Figure 11 shows a test that demonstrates the influence of dynamic effects on the contact force/moment. The held, rigid cylinder A was moved to hit the fixed, bottom cylinder B at a fixed angle θ with or without prior acceleration \mathbf{a} . When A hit B with no prior acceleration, A stuck upon contact under a rather large static friction force. Note that the short period where F_x dropped to zero and F_z became negative indicates a brief wobbling of A toward the $+z$ direction; the period ended when A was moved back to press along the $-x$ direction, and so on. Whereas, when A hit B with a prior acceleration \mathbf{a} , a larger normal force was generated, and the force of \mathbf{a} helped overcome the static friction so that A slid with a dynamic friction effect.

As presented, our method of modeling and computing haptic force and moment is general to all objects provided that their contact formation and configuration is known. In all of our experiments (with polyhedral and/or curved objects), modeling and computing haptic force and moment took an almost constant and instant time of approximately 12 to 16 μs —that is, had an

update rate of approximately 60 to 80 kHz—regardless of the objects' geometry. This was negligible compared to the time needed for contact detection (collision detection plus contact formation identification), which was in the order of kHz.

Future directions

An interesting future research task could be to integrate our work for haptic force and moment modeling based on accurate models of curved objects with fast collision detection and graphic rendering based on polygonal mesh models of the same objects (which could be of multiresolutions). Such a unified hybrid system could achieve even better results in terms of both accuracy and speed. Other directions include applying this work to virtual assembly tasks with more complex objects and extending this work to multilink articulated objects. ■

Acknowledgment

This work has been supported by the National Science Foundation grants IIS-9700412, EIA-0224423, and IIS-0328782.

References

1. R.J. Adams and B. Hannaford, "Stable Haptic Interaction with Virtual Environments," *IEEE Trans. Robotics and Automation*, vol. 15, no. 3, 1999, pp. 465-474.
2. B. Chang, *Real-Time Impulse-Based Simulation of Rigid Body Systems for Haptic Display*, doctoral dissertation, Dept. of Mechanical Engineering, Northwestern Univ., 2002.
3. P. Jimenez, F. Thomas, and C. Torras, "3D Collision Detection: A Survey," *Computers and Graphics*, vol. 25, no. 2, 2000, pp. 269-285.
4. D. Ruspini, K. Kolarov, and O. Khatib, "The Haptic Display of Complex Graphical Environments," *Proc. Siggraph 97*, ACM Press, 1997, pp. 345-352.
5. S. Hasegawa et al., "Real-Time Rigid Body Simulation Based on Volumetric Penalty Method," *Proc. 11th Symp. Haptic Interfaces for Virtual Environments and Teleoperator Systems*, IEEE Press, 2003, pp. 326-332.
6. Y. Kim et al., "Six Degree-of-Freedom Haptic Display Using

Localized Contact Computations," *Proc. 10th Symp. Haptic Interfaces for Virtual Environments and Teleoperator Systems*, IEEE Press, 2002, pp. 209-216.

7. B. Baraff, "Fast Contact Force Computation for Nonpenetrating Rigid Bodies," *Proc. Siggraph 94*, ACM Press, 1994, pp. 23-34.
8. D. Ruspini and O. Khatib, "Collision/Contact Models for Dynamic Simulation and Haptic Interaction," *Proc. 9th Int'l Symp. Robotics Research*, Springer-Verlag, 1999, pp. 185-194.
9. D. Stewart and J.C. Trinkle, "An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Press, 2000, pp. 162-170.
10. J. Xiao and X. Ji, "On Automatic Generation of High-Level Contact State Space," *Int'l J. Robotic Research (& Multimedia Extension)*, vol. 20, no. 7, July 2001, pp. 584-606.
11. L. Zhang and J. Xiao, "Derivation of Contact States from Geometric Models of Objects," *Proc. IEEE Int. Conf. Assembly and Task Planning*, IEEE CS Press, 1995, pp. 375-380.
12. Z. Zou and J. Xiao, "Tracking Minimum Distances between Curved Objects with Parametric Surfaces in Real Time," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, IEEE Press, 2003, pp. 2692-2698.



Qi Luo is a PhD candidate in the Department of Computer Science at the University of North Carolina, Charlotte. His research interests include haptic rendering, physics-based virtual reality, and robotics. Luo has a BS in physics from the University of Nankai, China and an MS in computer science from the University of North Carolina, Charlotte.



Jing Xiao is a professor in the Department of Computer Science at the University of North Carolina, Charlotte. Her research interests include robotics, physical simulation, haptics, intelligent systems, and optimization. Xiao has a PhD in computer, information, and control engineering from the University of Michigan.

Readers may contact the authors at the Dept. of Computer Science, Univ. of North Carolina, Charlotte, NC 28223; {qluo, xiao}@unc.edu.

PURPOSE The IEEE Computer Society is the world's largest association of computing professionals, and is the leading provider of technical information in the field.

MEMBERSHIP Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE The IEEE Computer Society's Web site, at www.computer.org, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

BOARD OF GOVERNORS

Term Expiring 2004: Jean M. Bacon, Ricardo Baeza-Yates, Deborah M. Cooper, George V. Cybenko, Harubisha Icbikawa, Thomas W. Williams, Yervant Zorian

Term Expiring 2005: Oscar N. Garcia, Mark A. Grant, Michel Israel, Stephen B. Seidman, Kathleen M. Swigger, Makoto Takizawa, Michael R. Williams

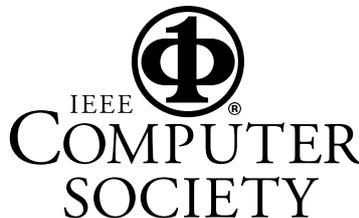
Term Expiring 2006: Mark Christensen, Alan Clements, Annie Combelles, Ann Gates, Susan Mengel, James W. Moore, Bill Schilit

Next Board Meeting: 11 Mar. 2005, Portland, OR

IEEE OFFICERS

President: ARTHUR W. WINSTON
President-Elect: W. CLEON ANDERSON
Past President: MICHAEL S. ADLER
Executive Director: DANIEL J. SENESE
Secretary: MOHAMED EL-HAWARY
Treasurer: PEDRO A. RAY

VP, Educational Activities: JAMES M. TIEN
VP, Pub. Services & Products: MICHAEL R. LIGHTNER
VP, Regional Activities: MARC T. APTER
VP, Standards Association: JAMES T. CARLO
VP, Technical Activities: RALPH W. WYNDRUM JR.
IEEE Division V Director: GENE F. HOFFNAGLE
IEEE Division VIII Director: JAMES D. ISAAK
President, IEEE-USA: JOHN W. STEADMAN



COMPUTER SOCIETY OFFICES
Headquarters Office

1730 Massachusetts Ave. NW
 Washington, DC 20036-1992
 Phone: +1 202 371 0101
 Fax: +1 202 728 9614
 E-mail: hq.ofc@computer.org

Publications Office

10662 Los Vaqueros Cir., PO Box 3014
 Los Alamitos, CA 90720-1314
 Phone: +1 714 821 8380
 E-mail: help@computer.org
Membership and Publication Orders:
 Phone: +1 800 272 6657
 Fax: +1 714 821 4641
 E-mail: help@computer.org

Asia/Pacific Office

Watanabe Building
 1-4-2 Minami-Aoyama, Minato-ku
 Tokyo 107-0062, Japan
 Phone: +81 3 3408 3118
 Fax: +81 3 3408 3553
 E-mail: tokyo.ofc@computer.org



EXECUTIVE COMMITTEE

President:

CARL K. CHANG*
 Computer Science Dept.
 Iowa State University
 Ames, IA 50011-1040
 Phone: +1 515 294 4377
 Fax: +1 515 294 0258
c.chang@computer.org

President-Elect: GERALD L. ENGEL*

Past President: STEPHEN L. DIAMOND*

VP, Educational Activities: MURALI VARANASI*
VP, Electronic Products and Services:

LOWELL G. JOHNSON (1ST VP)*

VP, Conferences and Tutorials:

CHRISTINA SCHOBERT†

VP, Chapters Activities:

RICHARD A. KEMMERER (2ND VP)*

VP, Publications: MICHAEL R. WILLIAMS*

VP, Standards Activities: JAMES W. MOORE*

VP, Technical Activities: YERVANT ZORIAN*

Secretary: OSCAR N. GARCIA*

Treasurer: RANGACHAR KASTURI†

2004-2005 IEEE Division V Director:

GENE F. HOFFNAGLE†

2003-2004 IEEE Division VIII Director:

JAMES D. ISAAK†

2004 IEEE Division VIII Director-Elect:

STEPHEN L. DIAMOND*

Computer Editor in Chief: DORIS L. CARVER†

Executive Director: DAVID W. HENNAGE†

* voting member of the Board of Governors
 † nonvoting member of the Board of Governors

EXECUTIVE STAFF

Executive Director: DAVID W. HENNAGE
Assoc. Executive Director: ANNE MARIE KELLY
Publisher: ANGELA BURGESS
Assistant Publisher: DICK PRICE
Director, Administration:
 VIOLET S. DOAN
Director, Information Technology & Services:
 ROBERT CARE