Robotica (2008) volume 26, pp. 513–524. © 2008 Cambridge University Press
doi:10.1017/S0263574708004207 Printed in the United Kingdom

Generalized penetration depth for penalty-based six-degree-of-freedom haptic rendering

Maxim Kolesnikov* and Miloš Žefran

Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607-7053.
Email: {mkoles2,mzefran}@uic.edu

(Received in Final Form: January 22, 2008. First published online: March 19, 2008)

SUMMARY

Existing penalty-based haptic rendering approaches are based on the penetration depth estimation in strictly translational sense and cannot properly take object rotation into account. We propose a new six-degree-of-freedom (6-DOF) haptic rendering algorithm which is based on determining the closest-point projection of the inadmissible configuration onto the set of admissible configurations. Energy is used to define a metric on the configuration space. Once the projection is found the 6-DOF wrench can be computed from the generalized penetration depth. The space is locally represented with exponential coordinates to make the algorithm more efficient. Examples compare the proposed algorithm with the existing approaches and show its advantages.

KEYWORDS: haptics; haptic rendering; six degrees of freedom; generalized penetration depth.

1. Introduction

A number of 6-degree-of-freedom (6-DOF) haptic devices have recently appeared on the market. They provide an improved touch-based human-machine interface, superior to the more common 3-DOF devices. However, the more realistic rendering of haptic scenes is achieved at the expense of increased complexity of calculations involved in force rendering algorithms.

Since 6-DOF haptic devices are relatively recent, 6-DOF force rendering algorithms are still an active research area; researchers are trying to improve the existing algorithms both in terms of realism as well as stability of haptic rendering. Thus, no 'standard' 6-DOF force rendering algorithm has emerged yet.[1,2]

Our work focuses on improving the realism of haptic rendering. The algorithm we propose is essentially a penalty-based method in which contact forces are computed as a function of current and desired configurations of a rigid body. Similarly to other approaches, the desired configuration is the configuration that takes a rigid body out of the collision. However, while most other approaches only consider translation, we compute the desired collision-free configuration based on the energy metric that properly takes into account the full rigid-body configuration. Subsequently, haptic feedback forces and torques are computed as in 3-DOF penalty-based methods using the computed generalized penetration depth.

The paper is organized as follows: related previous work is discussed in Section 1.1; in Section 2 we describe our haptic rendering algorithm; in Section 3 we discuss implementation-related issues; and in Section 4 we present simple numerical examples comparing the proposed ideas to the existing approaches. We conclude the paper with the discussion in Section 5 and outline possible future work in Section 6.

1.1. Related work

A popular approach to computing contact forces for the 3-DOF haptic rendering are penalty-based methods.[3] In this approach, once a collision between rigid bodies is detected, the penetration depth is estimated, where the penetration depth is defined as the shortest translation needed to separate the two bodies. Estimated penetration depth is used to compute the magnitude of the contact force, for example, as the sum of the proportional, integral, and differential terms with appropriate coefficients which can be changed to model various kinds of surfaces. The direction of the contact force is determined by the contact normal. This approach is also known as *direct rendering* and its main advantage is that the motion of the object entirely depends on the user of the haptic device, there is no need to consider the dynamics of the object. On the other hand, the method has its disadvantages, among them the pop-through artifacts and possible instability due to the force discontinuities.

Similar approach can be used for 6-DOF haptic rendering.[4] Contact torques in this case are computed as a cross product between the vector from the center of mass of the object to the contact point and the force vector. To increase the stability, this approach can be modified[5] by introducing a model of the object to which the calculated forces and torques are applied. This model interacts with the real haptic device through virtual coupling.[6]

Constraint-based methods[7,8] for contact force computation are based on the notion of a substitute virtual object. The virtual object never penetrates the virtual environment and its dynamics is constantly updated. Feedback force is calculated so that the user-controlled object is dragged toward the substitute virtual object. These methods can produce more accurate results at the expense of increased computational complexity. However, some recent results[9] promise to overcome some limitations of constraint-based methods.

Impulse-based techniques[10] implement the contact between two rigid bodies as a series of micro-collisions,

* Corresponding author: E-mail: mkoles2@uic.edu

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

and impulses are applied in order to prevent interpenetration. These techniques produce visually appealing but haptically unconvincing results.[10] Another approach is to combine penalty-based and impulse-based techniques.[10]

Another approach to 6-DOF haptic rendering is voxel sampling.[11] In this approach, the bodies in the scene are represented by voxels. The interaction forces and torques are computed for colliding voxels, subsequently summed up, and displayed to the user through a virtual coupling to filter out discontinuities.

The use of energy in haptic rendering has been considered in the extension of penalty methods[12] designed specifically for displaying haptic interaction between two textured models. Elastic penetration energy is considered, and haptic feedback forces and torques are defined as the gradient of elastic penetration energy.

Recently, attempts have been made[13] to extend the notion of penetration depth to take into account both translational and rotational motion to separate the intersecting objects. The underlying distance metric is defined as the length of the longest displacement vector over the corresponding vertices of the model between two configurations. Practical algorithms[14,15] for the computation of such generalized penetration depth have been proposed, but for convex models the generalized notion agrees with translation-based computations and thus shares the same shortcomings.

2. Proposed Algorithm

Our proposed algorithm falls into the class of penalty-based methods. The distinguishing feature of the new algorithm is a new method for computing the penetration depth during the collision between two rigid bodies. While the existing approaches essentially base the computation of the penetration depth only on the translational motion, we consider full rigid-body configuration (both translation and rotation). Formally, the penetration depth is described as a distance between two configurations of the rigid body, so it is necessary to define a metric on the configuration space to compute it. Motivated by the work in computational mechanics[16] we argue that the kinetic energy provides the appropriate metric and guarantees some desirable invariant properties of the computed penetration depth.

2.1. *Motivation*

To better understand the need for including the rotation in the penetration depth computation, consider two different haptic scenarios (Fig. 1.).

In both cases the cube controlled by the user through a haptic device lies on the plane. In case (a) the user is trying to push the cube into the surface and applies force F to the center of the contact plane whereas in case (b) the user is trying to rotate the cube into the surface so the force is applied at the edge of the cube. Existing penalty-based methods[4,5,10,17] would yield and send to the controller of the haptic device roughly the same contact force (albeit different torques)—the force that is perpendicular to the surface of the plane and is a function of the penetration depth defined in strictly translational sense. In reality, however, when the user applies pure torque to the object (with force components of the wrench being equal to zero), as in case (b), the computed contact force should not be the same as in (a); pure instantaneous torque applied to the object should correspond to the contact wrench that has only torque components. We thus argue that rather than basing the penetration depth computation on the translation only, a new approach is needed that can take into account full rigid-body configuration (both translation and rotation).

2.2. *Formal description of rigid body motion*

For kinematic and dynamic modeling we follow the approach based on configuration space description.[18,19] Let us consider a rigid body moving in space. Assume that there is a fixed inertial reference frame S and a reference frame T attached to the rigid body. In general, in order to describe the rigid body configuration one has to know both its position and orientation, that is the position and orientation of frame T with respect to frame S. All possible orientations of the rigid body form $SO(3)$, the *special orthogonal* group in $\mathbb{R}^3$ (in ref. [18]):

$$SO(3) = \left\{ R \in \mathbb{R}^{3\times 3} \,|\, RR^{\mathrm{T}} = I_{3\times 3}, \det R = 1 \right\}. \tag{1}$$

In a similar fashion in order to unambiguously represent all possible configurations (including both position and orientation) of the rigid body, members of $SE(3)$, the *special Euclidean* group in $\mathbb{R}^3$ (in ref. [18]), can be used:

$$SE(3) = \left\{ \begin{bmatrix} R & d \\ 0_{1\times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | R \in SO(3), d \in \mathbb{R}^3 \right\}. \tag{2}$$

It can be shown that both $SO(3)$ and $SE(3)$ are indeed groups under the operation of matrix multiplication[18] and they are smooth three- and six-dimensional manifolds, respectively, and therefore Lie groups.[20]

At the identity element of a Lie group the tangent space has the structure of a Lie algebra which completely captures the local structure of the group. The Lie algebra of $SO(3)$ is given by ref. [18]

$$so(3) = \{\Omega \in \mathbb{R}^{3\times 3} | \Omega^T = -\Omega\}. \tag{3}$$

Each element $\Omega \in so(3)$ can be identified with a vector $\omega \in \mathbb{R}^3$ so that $\Omega x = \omega \times x$, where the symbol $\times$ denotes the cross product of two vectors. We will also write $\Omega = \widehat{\omega}$.

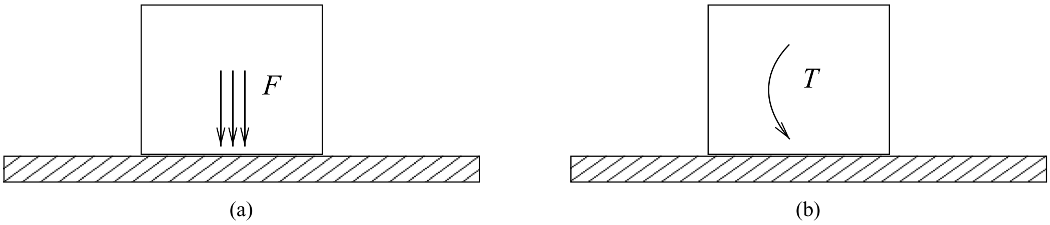

Fig. 1. When the penetration depth for haptic rendering is based on the translation only, forces in cases (a) and (b) are equal.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

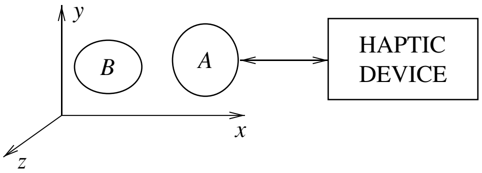

Fig. 2. Two rigid bodies in a virtual environment.

Similarly, the Lie algebra of $SE(3)$ is given by ref. [18]

$$se(3) = \left\{ \begin{bmatrix} \Omega & v \\ 0_{1\times 3} & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} | \Omega \in so(3), v \in \mathbb{R}^3 \right\}. \quad (4)$$

An element of $se(3)$ is referred to as a *twist*. Each twist can be identified with a vector $\xi = [\, v^{\mathrm{T}}\ \omega^{\mathrm{T}}\,]^{\mathrm{T}} \in \mathbb{R}^6$ where $\Omega = \widehat{\omega}$. The vector $\xi \in \mathbb{R}^6$ represents the *twist coordinates* of the twist Ξ, and we write $\Xi = \widehat{\xi}$.

Every possible rigid body configuration is represented by an element g of $SE(3)$ which in turn can be locally described by a 6×1 vector ξ of *exponential coordinates* through the (matrix) exponential mapping,[18] $g = \exp(\widehat{\xi}), \widehat{\xi} \in se(3)$.

2.3. Penetration depth computation

Consider two rigid bodies A and B moving relative to each other (Fig. 2). The rigid body A is controlled by the user through the haptic device and the rigid body B is an obstacle in the virtual environment. Let us denote the configuration space of the system by Q. To keep things simple we only consider the relative position of object A with respect to object B, and we assume that there are no boundary surfaces so that the configuration space is $Q = SE(3)$. Let $A(q)$ be the subset of $\mathbb{R}^3$ occupied by the object A at configuration $q \in Q$. Then the admissible configuration set C is given by a set of all possible configurations $q \in Q$ for which the two bodies do not collide:

$$C = \{q \in Q | A(q) \cap B = \emptyset\}. \quad (5)$$

Consider the situation when the body A is in collision with the body B. We denote this inadmissible configuration by p. Assuming that the admissible set $C \subset Q$ is nonempty we can define the *distance function*[16] as

$$d_C(p) = \min\{d(p, q) \mid q \in C\}, \quad (6)$$

which returns the distance between any inadmissible configuration p and the admissible set C. Here the function $d(p, q)$ is the *distance metric function*[21] between configurations p and q.

For every inadmissible configuration p we can find an admissible configuration q such that $q \in P_C(p)$, where

$$P_C(p) = \{q \in \overline{C} \mid d_C(p) = d(p, q)\}. \quad (7)$$

We use $\overline{C}$ to denote the closure of C. The configuration q is called a *closest-point projection*[16] of p onto C.

Naturally, the question comes up of what distance metric function to choose for the computation of $d(p, q)$. It is important for the metric not to be affected by the scaling of the objects. This is important because if, for example, an object P is a scaled version of an object Q, they would look identical to the user in a haptic simulation since the scale used for the graphic display of the virtual environment is not absolute. Therefore, the computed closest-point projections q_P and q_Q should also look the same, with the translational part of one being the scaled version of the translational part of the other. This condition rules out $d(p, q) = d(I_{4\times 4}, p^{-1}q) = ||\xi_{p^{-1}q}||_2$ as a possible solution, where $\xi_{p^{-1}q}$ represents the exponential coordinates of the configuration $p^{-1}q$. An attractive solution is to define the distance metric as the kinetic energy needed to move between configurations p and q in a unit of time:

$$d(p, q) = T = \widetilde{V}^{\mathrm{T}} M \widetilde{V}, \quad (8)$$

where $\widetilde{V}$ is defined below and roughly corresponds to the twist coordinates of the body velocity of the rigid body A moving from p to q, while M is the generalized inertia matrix. To compute the body velocity $\widetilde{V}$ represented in the twist coordinates we use the fact that if $g(t) \in SE(3)$ describes the motion of a rigid body, then the body velocity $\widehat{V} \in se(3)$ of the rigid body is[18]

$$\widehat{V} = g^{-1}\frac{dg}{dt}. \quad (9)$$

To compute $\widetilde{V}$, take g to be the configuration located at the midpoint between p and q, that is

$$g = p\ \exp\left(\left[\frac{\xi_{p^{-1}q}}{2}\right]^{\wedge}\right). \quad (10)$$

The derivative of g can be approximated by

$$\frac{dg}{dt} = \frac{q - p}{\Delta t}. \quad (11)$$

The time interval Δt it takes to get from the inadmissible configuration p to an admissible configuration q is assumed to be the same for each possible configuration q considered. Therefore, without a loss in generality we can assume that $\Delta t = 1$ so that $dg/dt \approx q - p$. We thus have the final expression:

$$\left[\widetilde{V}\right]^{\wedge} = \exp\left(\left[-\frac{\xi_{p^{-1}q}}{2}\right]^{\wedge}\right) p^{-1}(q - p), \quad (12)$$

where $\widetilde{V}$ are the twist coordinates of $\left[\widetilde{V}\right]^{\wedge}$.

Recall that the generalized inertia matrix M in the case where the body reference frame is placed at the center of mass of a rigid body is given by

$$M = \begin{bmatrix} mI_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & H \end{bmatrix}, \quad (13)$$

where m is the mass of the object and H is the inertia tensor.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

The distance metric defined through Eqs. (8) and (12) is insensitive to the scaling of the objects. Another important property of this distance metric is its *bi-invariance*: no matter how the inertial frame and the body reference frame are chosen, the distance metric $d(p, q)$ remains the same. This is due to the nature of the kinetic energy. Formally, the bi-invariance means that

$$d(p, q) = d(ap, aq) = d(pb, qb), \tag{14}$$

where $a, b \in SE(3)$. The bi-invariance of the distance metric is sufficient for the metric to be *well-defined*.[21] Therefore, there is no need to stay in the neighborhood of the identity element of $SE(3)$ when measuring the distance between two configurations. It is also worth mentioning that the mass properties of the object will not affect the metric, but it does depend on the shape of the object. The latter can be seen as an advantage over the approach in ref. [4], where the reaction wrench is independent of the shape of the object.

Note that using the metric in Eq. (8), out of all possible motions between the current configuration p and configurations in the admissible set C, the motion between p and its closest-point projection q requires the least amount of kinetic energy. The above defined closest-point projection is therefore *energy optimal*.

2.4. *Wrench computation*

Suppose we determine the optimal admissible configuration q as the closest-point projection onto the admissible set C in accordance with Eq. (7). Then the penalty-based approach for the wrench computation can be used. First we compute the vector ε of the differences between the twist coordinates of the (current) inadmissible configuration p and the optimal admissible configuration q. Note that in order for the result to be independent of the chosen inertial reference frame, we compute the difference at the identity element of $SE(3)$:

$$\varepsilon = \xi_{p^{-1}q} - \xi_{p^{-1}p} = \xi_{p^{-1}q}. \tag{15}$$

One can think of the vector ε as the generalized penetration depth. It is equal to zero if and only if $p = q$. This can only happen when $p \in C$ which means that the rigid body is not in collision.

Once the penetration depth has been computed, any appropriate scheme can be used to compute the wrench. Motivated by the existing approaches, we use the proportional-integral-derivative (PID) control law. Accordingly, the wrench $W_T(t) = [\, f^{\mathrm{T}}(t) \quad \tau^{\mathrm{T}}(t)\,]^{\mathrm{T}}$ at each time instance is computed as

$$W_T(t) = K_P \varepsilon(t) + K_I \int_0^t \varepsilon(\tau) d\tau + K_D \frac{d\varepsilon(t)}{dt}, \tag{16}$$

where K_P, K_I, K_D are 6×6 matrices. In fact, the matrices K_P and K_D have the physical interpretation as the stiffness and damping matrices, respectively. Even though the stiffness matrix K_P is in general asymmetric,[22] for the sake of simplicity one can assume that it is symmetric and diagonal. It can be therefore written as $K_P = k_P I_{6\times 6}$, so that a single stiffness parameter k_P is needed. The same simplification can be made for matrices K_I and K_D.

The wrench $W_T(t)$ is expressed in the body coordinate frame. The same wrench expressed in the global coordinate frame is $W_S(t) = \mathrm{Ad}^{\mathrm{T}}_{p^{-1}} W_T(t)$, where for $p = (R, d) \in SE(3)$, the *adjoint transformation* $\mathrm{Ad}_p : \mathbb{R}^6 \to \mathbb{R}^6$ is

$$\mathrm{Ad}_p = \begin{bmatrix} R & \widehat{d}R \\ 0 & R \end{bmatrix}.$$

The wrench $W_S(t)$ is the nominal wrench sent to the haptic device.

3. Implementation

The procedure for computation of the six-dimensional wrench is described by Algorithm 1. The data flow diagram for the algorithm is given in Fig. 3. In order to find the closest-point projection onto the admissible set C in line 4 of Algorithm 1, it is necessary to minimize the function $d(p, q)$. There are two different ways of approaching this problem: numerical optimization and analytical methods.

Algorithm 1 Haptic rendering algorithm.

```
1:  loop
2:    if $A$ and $B$ in collision then
3:      Let $p$ be the current inadmissible configuration.
4:      Using the proposed kinetic energy metric, find the
        admissible configuration $q = \arg\min_{r\in C} d(p, r)$.
5:      Compute the configuration matrix $p^{-1}q$ corres-
        ponding to the relative displacement between $p$ and $q$.
6:      Compute the difference vector $\varepsilon = f_1(p, q)$ using Eq. (15).
7:      Compute the body wrench $W_T = f_2(\varepsilon)$ using Eq.
        (16) and find the corresponding spatial wrench $W_S$.
8:      Set the wrench $W_S$ to be the nominal input for the
        controller of the haptic device.
9:    end if
10: end loop
```

Generically, assuming fast refresh rates, either vertex-face or edge-edge collision configurations are possible.[23] We will describe the computations for the vertex-face collision and

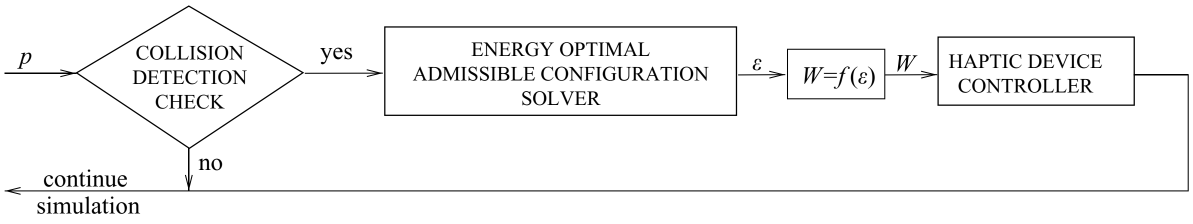
p
COLLISION DETECTION CHECK
yes
no
ENERGY OPTIMAL ADMISSIBLE CONFIGURATION SOLVER
ε
W=f(ε)
W
HAPTIC DEVICE CONTROLLER
continue simulation

Fig. 3. Data flow diagram for the 6-DOF wrench computation.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

subsequently show that the edge-edge collision can be dealt with in a similar fashion.

3.1. Computing the distance using numerical optimization

There are many well-known optimization methods that can be applied to finding the optimal (minimum distance) configuration of the rigid body. The main requirement is that the optimization procedure is as fast and efficient as possible. If the minimization is done directly in $SE(3)$ then a full 4×4 configuration matrix must be searched for. However, since not every 4×4 matrix is a member of $SE(3)$, we can use this information to lower the number of unknown optimization parameters. In particular, we can represent the configuration space $SE(3)$ in the neighborhood of the current inadmissible configuration p with exponential coordinates which in turn correspond to vectors in $\mathbb{R}^6$. Also, it is convenient to start the optimization with the last recorded admissible configuration as an initial guess.

The main limitation of the proposed algorithm is its computational cost: even simple numerical examples take around 4 sec to run in Matlab on a 1 GHz CPU. While this time can be reduced through compilation it is clear that the optimization-based approach can be quite computationally demanding. Therefore, even though the approach is general and simple to implement, the computational complexity of applying direct numerical optimization in complex virtual environments might be prohibitively high for real-time implementation.

3.2. Analytical solution in 2D

As an alternative to the numerical optimization we propose an analytical approximation of the solution that is fairly accurate and can be computed at sufficiently high rates. We first describe the method for a two-dimensional case, where the analytical solution is exact. Then we discuss how to use these results in a general three-dimensional case.

In a two-dimensional case a planar body can only move on a plane and rotate around the axis perpendicular to the plane. Let us assume that the body is a convex polygon. Let us also assume that only one vertex of the body is in collision. This is virtually always the case for real-time simulations with refresh rates of several hundred hertz. After the collision is detected the body resides in some inadmissible configuration p. Such situation is depicted in Fig. 4, where the region below the x-axis coincides with the planar obstacle. Here, T is the body reference frame placed at the center of mass of the body, h is the y-coordinate of the reference frame T, and φ is the angle between the reference frame T and the global reference frame.

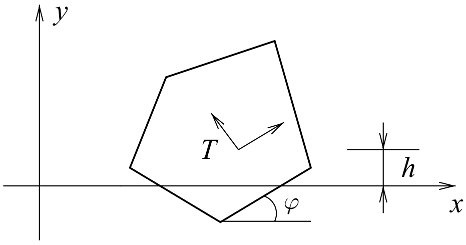

Fig. 4. Inadmissible configuration of a planar body after collision.

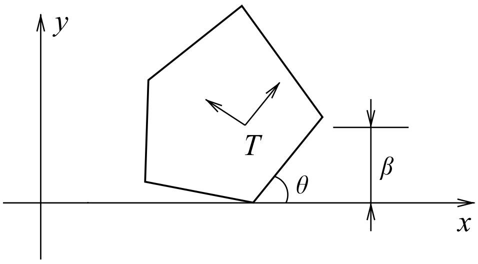

Fig. 5. Minimum energy admissible configuration of a planar body.

We are looking for the admissible configuration $q \in P_C(p)$ of the body that is a closest-point projection of configuration p onto the admissible set C. Such a configuration q minimizes the distance metric given by Eq. (8). Configuration q (Fig. 5) can be parametrized by the y-coordinate β of the body reference frame T and the angle θ between the reference frame T and the main spatial reference frame. Using energy considerations it can be shown that the x-coordinate of the reference frame T corresponding to the optimal configuration q is the same as the x-coordinate of p.

In this general setup the body velocity V can be evaluated according to the expression in Eq. (12), and

$$V = \begin{bmatrix} (\beta - h)\sin\left(\dfrac{\theta+\varphi}{2}\right) \\ (\beta - h)\cos\left(\dfrac{\theta+\varphi}{2}\right) \\ 0 \\ 0 \\ 0 \\ 2\sin\left(\dfrac{\theta-\varphi}{2}\right) \end{bmatrix}. \tag{17}$$

Then, according to Eq. (8),

$$T = m(\beta - h)^2 + 4H_{3,3}\sin^2\left(\frac{\theta-\varphi}{2}\right), \tag{18}$$

where $H_{3,3}$ is the $(3, 3)$ element of the inertia tensor matrix H. Given parameters h and φ, we can find β and θ in the admissible set such that the expression in Eq. (18) is minimized. Given β and θ, the desired configuration q is then computed as

$$q = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & p_{1,4} \\ \sin\theta & \cos\theta & 0 & \beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{19}$$

where $p_{1,4}$ is the $(1, 4)$ element of the original configuration matrix p.

Using energy considerations it can be shown that in the optimal configuration, the y-coordinate of the vertex v is

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

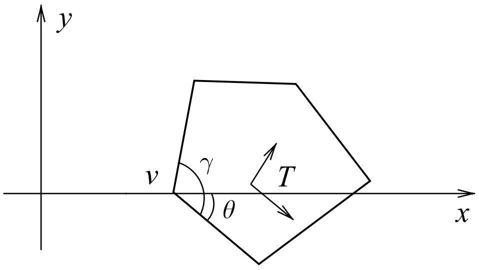

Fig. 6. An example of a situation that has to be accounted for in optimization constraints.

equal 0. This results in

$$\beta = -v_x \sin\theta - v_y \cos\theta, \tag{20}$$

where v_x and v_y are x- and y-coordinates of the vertex v in the body reference frame, respectively. To prevent situations like the one depicted in Fig. 6, we also need to satisfy the constraint

$$0 \leq \theta \leq \pi - \gamma, \tag{21}$$

where γ is the internal angle of the planar body associated with the vertex v. This constraint will be used later to check whether the obtained solution is in the admissible configuration set.

Using Eqs. (18) and (20), our optimization function is

$$J = m(-v_x \sin\theta - v_y \cos\theta - h)^2 + 4H_{3,3} \sin^2\left(\frac{\theta - \varphi}{2}\right). \tag{22}$$

To find the critical value $\widetilde{\theta}$ of the parameter θ minimizing the functional given by Eq. (22), the following equation needs to be solved:

$$\begin{aligned} \left.\frac{\partial J}{\partial \theta}\right|_{\theta=\widetilde{\theta}} &= m\left(v_x^2 - v_y^2\right)\sin 2\theta + 2mv_xv_y\cos 2\theta \\ &\quad + 2mh(v_x\cos\theta - v_y\sin\theta) \\ &\quad + 2H_{3,3}\sin(\theta - \varphi) = 0. \end{aligned} \tag{23}$$

It can be shown that solving Eq. (23) for $\widetilde{\theta}$ results in

$$\widetilde{\theta} = 2\arctan x, \tag{24}$$

where x is a solution of a quartic equation

$$A_1x^4 + B_1x^3 + C_1x^2 + D_1x + E_1 = 0, \tag{25}$$

where

$$A_1 = mv_xv_y - mv_xh + H_{3,3}\sin\varphi, \tag{26}$$

$$B_1 = 2\left(H_{3,3}\cos\varphi - mhv_y - mv_x^2 + mv_y^2\right), \tag{27}$$

$$C_1 = -6mv_xv_y, \tag{28}$$

$$D_1 = 2\left(H_{3,3}\cos\varphi - mhv_y + mv_x^2 - mv_y^2\right), \tag{29}$$

$$E_1 = mv_xv_y + mv_xh - H_{3,3}\sin\varphi. \tag{30}$$

Closed form solutions of quartic equations can be obtained using for example the classic Ferrari's method[24] which is fast and reliable.

After solving Eq. (25) for x and obtaining the critical value $\widetilde{\theta}$ of θ the constraint described by Eq. (21) has to be checked and the value $T_{\theta=\widetilde{\theta}}$ of the energy function in Eq. (18) has to be computed. Also, the border values $T_{\theta=0}$ and $T_{\theta=\pi-\gamma}$ of the energy function have to be computed for $\theta = 0$ and $\theta = \pi - \gamma$. If the obtained $\widetilde{\theta}$ satisfies the constraint, then the optimal value θ_0 of θ is

$$\theta_0 = \arg\min_{\theta}\{T_{\theta=0}, T_{\theta=\widetilde{\theta}}, T_{\theta=\pi-\gamma}\}. \tag{31}$$

If the obtained $\widetilde{\theta}$ does not satisfy the constraint described by Eq. (21), then

$$\theta_0 = \arg\min_{\theta}\{T_{\theta=0}, T_{\theta=\pi-\gamma}\}. \tag{32}$$

From here, the optimal values of $\beta_{\theta=\theta_0}$ and $q_{\theta=\theta_0}$ could be obtained as shown in Eqs. (20) and (19).

3.3. *Approximation of analytical solution in 3D*

We will obtain a solution for the 3D case by combining solutions for 2D cases. If we have a 3D virtual environment with a user-controlled convex rigid body A, then we can apply the 2D technique described in Section 3.2 to any cross section of the rigid body containing the vertex v in collision.

Without loss of generality let us arrange the axes X, Y, and Z so that the plane XZ is the surface of the obstacle and some vertex v collided with it. The projection of the rigid body on the XY-plane thus corresponds to Fig. 4.

Now consider the family of planes $\Pi = \{P_\delta \mid \delta \in [0, \pi)\}$ obtained by rotating the XY plane around the line parallel to the Y-axis and containing the vertex v. Each plane P_δ of the family is thus parametrized by its rotation angle, $\delta = \angle(P_\delta, XY)$. Figure 7 shows an example of two distinct planes P_{δ_1} and P_{δ_2} belonging to the family Π.

Each δ generates a planar cross-section $P_\delta \cap A$. Considering this cross-section to be a 2D body, it is possible to compute the optimal solution of the rigid body constrained to P_δ using the procedure described in Section 3.2. The plane

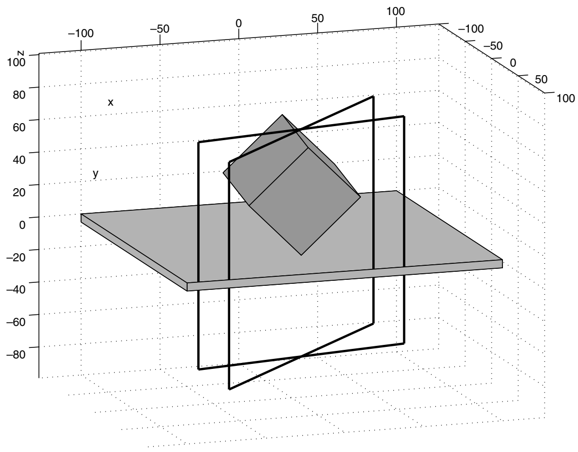

Fig. 7. Example of two bisecting planes P_{δ_1} and P_{δ_2} in family Π.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

P_δ passing through the vertex v is given by the equation

$$P_\delta(v) = A_2 x + C_2 z + D_2 = 0, \tag{33}$$

where

$$A_2 = \tan\delta, \tag{34}$$

$$C_2 = 1, \tag{35}$$

$$D_2 = -v_x \tan\delta - v_z, \tag{36}$$

and v_x and v_z are x- and z-coordinates, respectively, of the vertex v in the global reference frame.

In order to follow the procedure in Section 3.2, the internal angle γ of the planar cross-section of the rigid body needs to be computed. Let us assume that the rigid body A is described by a set of triangles $\Gamma_1, \Gamma_2, \ldots, \Gamma_m$ (as it is common in computer graphics) and consider a sequence of points $N_v = \{v_1, v_2, \ldots, v_n\}$ such that $\forall i \in N_v$ a triplet of points $\{v, v_i, v_{i+1}\}$ (where $v_{n+1} \equiv v_1$ by convention) describes a triangular face Γ_i adjacent to the vertex v. Given a plane P_δ, the angle γ can be then computed using Algorithm 2.

Algorithm 2 Algorithm for computing the cross-section angle γ.

```
if $P_\delta$ coincides with a face $\Gamma_i$ then
    $\vec{w}_1 = v_i - v$
    $\vec{w}_2 = v_{i+1} - v$
else
    if $P_\delta$ passes through two distinct edges $(v, v_i)$ and $(v, v_j)$ then
        $\vec{w}_1 = v_i - v$
        $\vec{w}_2 = v_j - v$
    else
        if $P_\delta$ passes through an edge $(v, v_i)$ and bisects a face $\Gamma_j$ then
            $\vec{w}_1 = v_i - v$
            $\vec{w}_2 = v_j + s(v_{j+1} - v_j) - v$ where $s = \frac{A_2 v_{j,x} + C_2 v_{j,z} + D_2}{A_2 v_{j,x} + C_2 v_{j,z} - A_2 v_{j+1,x} - C_2 v_{j+1,z}}$
        else
            if $P_\delta$ bisects two distinct faces $\Gamma_i$ and $\Gamma_j$ then
                $\vec{w}_1 = v_i + s(v_{i+1} - v_i) - v$ where $s = \frac{A_2 v_{i,x} + C_2 v_{i,z} + D_2}{A_2 v_{i,x} + C_2 v_{i,z} - A_2 v_{i+1,x} - C_2 v_{i+1,z}}$
                $\vec{w}_2 = v_j + s(v_{j+1} - v_j) - v$ where $s = \frac{A_2 v_{j,x} + C_2 v_{j,z} + D_2}{A_2 v_{j,x} + C_2 v_{j,z} - A_2 v_{j+1,x} - C_2 v_{j+1,z}}$
            end if
        end if
    end if
end if
$\gamma = \arccos \frac{\vec{w}_1 \cdot \vec{w}_2}{|\vec{w}_1||\vec{W}_2|}$
```

Next, we discuss how the 2D solutions for $\delta \in [0, \pi)$ can be combined to form a general 3D solution. As we have seen, for a given object with the known mass, inertia matrix, center of mass, and coordinates of the vertices, a 2D problem can be solved for each value of the angle δ (and the bisecting plane P_δ) using the procedure in Section 3.2. Let us denote the obtained solution as

$$G_{2D}(\delta) = \begin{bmatrix} \beta_\delta \\ \theta_\delta \end{bmatrix}. \tag{37}$$

Recall that θ_δ is the optimal angle of rotation in the plane P_δ. It is easy to see that the corresponding axis of rotation ω_δ expressed as a vector in 3D for a particular value of δ can be determined as $\omega_\delta = [\,\sin\delta \quad 0 \quad \cos\delta\,]^{\mathrm{T}}$.

Solution in Eq. (37), obtained in 2D, can be expressed in 3D by adding the information about the rotation axis ω_δ. Let us denote

$$G_{3D}(\delta, G_{2D}(\delta)) = \begin{bmatrix} \beta_\delta \\ \theta_\delta \\ \omega_\delta \end{bmatrix}. \tag{38}$$

For each $G_{3D}(\delta, G_{2D}(\delta))$ there is the corresponding 3D configuration $q(G_{3D}(\delta, G_{2D}(\delta)))$ and the distance metric $d(p, q(G_{3D}(\delta, G_{2D}(\delta))))$. Then the optimal solution $[\,\beta_{\delta_0}\ \omega_{\delta_0}\ \theta_{\delta_0}\,]^{\mathrm{T}}$ which describes the closest point projection of p can be determined by

$$\delta_0 = \arg\min_{\delta \in [0,\pi)} d(p, q(G_{3D}(\delta, G_{2D}(\delta)))). \tag{39}$$

In practice, δ_0 can be found by appropriately discretizing the interval $[0, \pi)$ and finding the maximum over the resulting discrete set Σ:

$$\delta_0 \approx \arg\min_{\delta \in \Sigma} d(p, q(G_{3D}(\delta, G_{2D}(\delta)))). \tag{40}$$

A convenient choice for Σ is $\Sigma = \left\{\frac{n\pi}{N_\delta} \middle| n = 0, 1, \ldots, N_\delta - 1\right\}$ where N_δ is the number of discretization points.

3.4. Edge-edge collision

We next show that the edge-edge collision (Fig. 8) can be dealt with similarly to the vertex-face collision considered earlier. Assume that the edges in contact are a and b, and that they belong to rigid bodies A and B, respectively. To simplify the problem consider the edges to be infinite lines in space.

It can be seen that there are only two rotational degrees of freedom that are necessary to bring the rigid body A out of collision. To illustrate this, consider the plane Q such that $a \subset Q$ and $b || Q$ (the singular case when the two edges

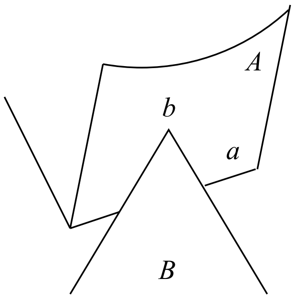

Fig. 8. A general edge-edge collision configuration.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

coincide can be treated as two vertex-face contacts). It is then obvious that the rotation of A in the plane Q is not energy optimal. A similar argument shows that only the translation in the direction perpendicular to Q should be used to compute the closest-point projection.

Upon closer inspection, the parameters that describe the closest-point projection in the edge-edge case are thus identical to the vertex-face collision. Due to the same parametrization the two problems thus have identical solution, so the edge-edge collision can be treated similarly to the vertex-face case. In particular, taking the bisection of the object A with a plane P perpendicular to Q and containing the common normal of the edges a and b, we obtain exactly the configuration described by Fig. 4. The closest-point projection restricted to P can be thus computed as described in Section 3.2. The 3D solution can be subsequently obtained by rotating the plane P along the common normal of the edges a and b and combining the 2D solutions as outlined in Section 3.3.

3.5. Computational performance of analytical approximation

One of the most important requirements for any haptic rendering algorithm is its ability to provide results in the specified time interval. In haptics, 1 kHz is the *de facto* standard refresh rate. Studies have shown that the absolutely minimal acceptable haptic refresh rate is 500 Hz.[25]

We have analyzed the performance of the proposed analytical method on a 1 GHz CPU. For the purpose of this analysis the proposed method can be divided into three stages:

1. Computation of a 2D solution. The query rate for this stage only was measured to be about $f_1 = 60$ kHz. It is roughly constant for all possible cases.
2. Computation of the angle γ of each planar cross-section. The query rate for this stage in the worst case scenario corresponding to lines 14–15 in Algorithm 2 was measured to be at least $f_2 = 226$ kHz. For other cases it was greater than this value.
3. The procedure for combining the family of 2D solutions into a 3D optimal solution. The query rate for this stage is very high compared to the other two stages, so this stage can generally be neglected.

Query rate for stages 1 and 2 combined is therefore

$$f = \left(f_1^{-1} + f_2^{-1}\right)^{-1} = 47\ \text{kHz}. \tag{41}$$

The computation in stages 1 and 2 should be performed N_δ times, so, taking into account the overhead in stage 3, the effective query rate is

$$f_0 \leq \frac{f}{N_\delta}. \tag{42}$$

For realistic haptic rendering we need to make sure that $f_0 \geq 1$ kHz, therefore in our case $N_\delta \leq 47$. This would give us the step size $\Delta\delta \geq 3.8^\circ$ which gives sufficient degree of accuracy. It is worth mentioning that the computations in stages 1 and 2 can be performed in parallel. In particular, the algorithm can easily take the advantage of the modern multi-core processors so we argue that it is a feasible alternative to the haptic rendering algorithms currently in use.

4. Examples

We illustrate the proposed methodology with two examples. The first example shows how the closest-point projection can be computed for a typical collision configuration. In the second example, the trajectories of a rigid body are computed under the proposed haptic rendering algorithm, and under a traditional translation-based haptic rendering algorithm. It is shown that the two algorithms produce significantly different torque profiles.

4.1. Example: Closest-point projection calculation

Consider a virtual environment consisting of a cube controlled by the user's haptic device and an obstacle in the form of a plane. In this particular example the cube has sides of length 50 cm and it can move freely in the half-space $\{y > 0\}$. The mass m of the cube is 1 kg (note that the choice of the mass does not affect the computation). Suppose the user applies a pure torque trying to rotate the cube into the plane. After the collision detection routine has detected the interpenetration of the cube and the plane we have a situation like the one depicted in Fig. 9(a). This configuration

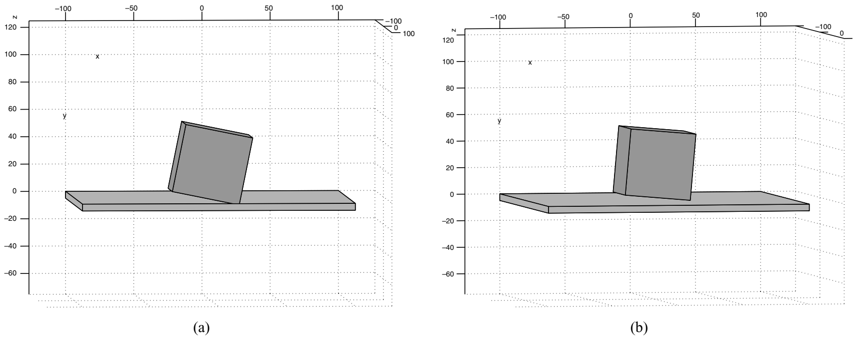

Fig. 9. Example: (a) a cube is rotated into the plane by the user, (b) the closest admissible configuration (in the energy optimal sense).

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

is described by a configuration matrix

$$p = \begin{bmatrix} 0.9801 & 0.1987 & 0 & 0 \\ -0.1987 & 0.9801 & 0 & 25.0000 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \quad (43)$$

The exponential coordinates of p are

$$\xi_p = \begin{bmatrix} -2.5000 \\ 24.9166 \\ 0 \\ 0 \\ 0 \\ -0.2000 \end{bmatrix}. \quad (44)$$

The closest-point projection on the admissible set C described by Eqs. (6), (7) and (8) was computed through numerical optimization. The exponential coordinates of the energy optimal configuration q are

$$\xi_q = \begin{bmatrix} -1.1737 \\ 27.0543 \\ 0 \\ 0 \\ 0 \\ -0.0867 \end{bmatrix}. \quad (45)$$

The corresponding configuration matrix is

$$q = \begin{bmatrix} 0.9962 & 0.0866 & 0 & 0 \\ -0.0866 & 0.9962 & 0 & 27.0712 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \quad (46)$$

The minimum of the energy function in Eq. (8) is found to be $T_{\min} = 9.6315$. The difference vector can be computed as

$$\varepsilon = \xi_{p^{-1}q} = \begin{bmatrix} -0.2959 \\ 2.0510 \\ 0 \\ 0 \\ 0 \\ 0.1133 \end{bmatrix}. \quad (47)$$

In this case we used the conjugate gradient method[26] for the numerical optimization. The configuration space $SE(3)$ was represented with the exponential coordinates so that the search was performed on the space $\mathbb{R}^6$. The vector of exponential coordinates ξ_{q_0} corresponding to the last recorded admissible configuration q_0 was chosen as the initial value for the optimization procedure:

$$\xi_{q_0} = \begin{bmatrix} 0 \\ 25.0000 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (48)$$

$$q_0 = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 25.0000 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \quad (49)$$

Analysis of the solution in Eq. (47) reveals that the wrench needed to bring the cube from configuration p to configuration q consists of forces in both Y- and X-directions as well as a torque around the Z-axis. Comparing the computed wrench with the one computed by the traditional penalty-based computational approach, we see that the latter would have only the force in the Y-direction with a greater magnitude than in our solution (since the penetration depth in configuration p is around 4.5 mm) as well as the torque computed as a cross product between the vector from the center of mass of the object to the contact point and the force vector. The wrench computed using our proposed method is thus quite different from the one computed by the existing penalty-based methods. The closest admissible configuration in the energy optimal sense is shown in Fig. 9(b).

Let us look at the same example but with dimensions of the cube and all the coordinates of the virtual environment scaled by a factor of 10. Thus, the inadmissible configuration matrix

$$p = \begin{bmatrix} 0.9801 & 0.1987 & 0 & 0 \\ -0.1987 & 0.9801 & 0 & 250.0000 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \quad (50)$$

In this case our proposed procedure yields the energy optimal configuration matrix

$$q = \begin{bmatrix} 0.9962 & 0.0866 & 0 & 0 \\ -0.0866 & 0.9962 & 0 & 270.7122 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \quad (51)$$

These configurations are depicted in Fig. 10. The obtained solution is exactly the same as in the previous case except that all the translational parameters are scaled by a factor of 10. Our proposed method handles these situations appropriately because the rigid body configuration it finds is the one that requires the minimum amount of energy.

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

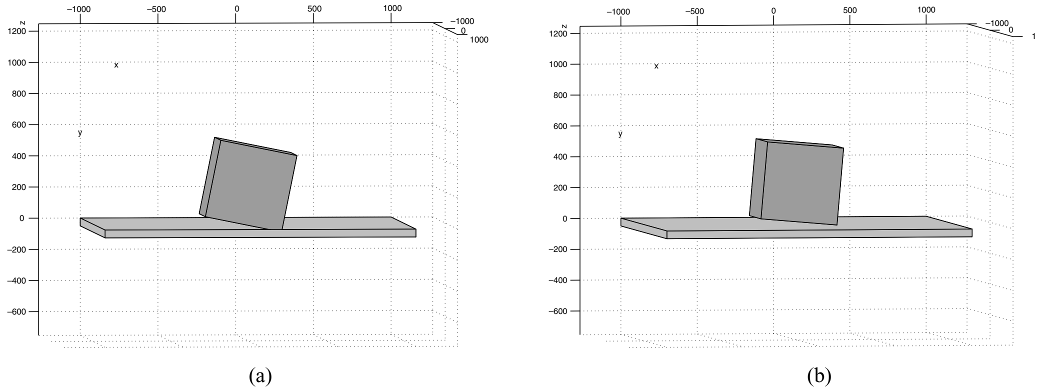

Fig. 10. Example: (a) a cube is rotated into the plane by the user, (b) the closest admissible configuration (in the energy optimal sense). The virtual environment is scaled by a factor of 10.

4.2. Example: Dynamic simulation

To illustrate the differences in the results obtained using our proposed method and the traditional penalty-based approaches we study the trajectories during a collision of two rigid bodies. Consider a virtual environment as in Example 4.1, where the cube can move freely in the half-space $\{y > 0\}$, the other half-space representing an obstacle. The initial inadmissible configuration is described by the configuration matrix

$$p = \begin{bmatrix} 0.6061 & -0.7954 & 0 & 0 \\ 0.7954 & 0.6061 & 0 & 34.1817 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \tag{52}$$

The exponential coordinates of p are

$$\xi_p = \begin{bmatrix} 15.7174 \\ 31.7380 \\ 0 \\ 0 \\ 0 \\ 0.9196 \end{bmatrix}. \tag{53}$$

In the example we consider the case when the cube is initially rotating with a constant body velocity

$$V = \begin{bmatrix} 0 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0.2 \end{bmatrix}. \tag{54}$$

Then, at a certain point it collides with the obstacle plane and the haptic rendering algorithm is run to compute the forces and torques resulting from the collision. In this case we used the proposed analytical approximation method to find the optimal minimum-energy admissible configuration. The computed forces and torques are applied to the cube and affect its motion until it gets out of the collision, after which it continues to move with a constant velocity. The simulation was run with a sampling period of $\Delta T = 0.05$sec for $N_t = 120$ steps.

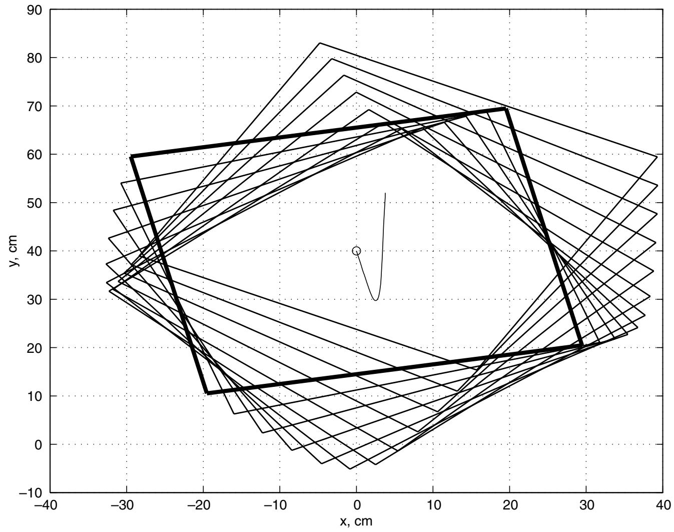

Fig. 11. The trajectory of the rigid body and its center of mass (solid line) projected on the XY-plane using the proposed energy-optimal method. The initial configuration of the body is shown in bold and the initial position of its center of mass is represented with a circle.

We ran our proposed method side by side with the traditional penalty-based approach suitable for 6-DOF haptic rendering[4] (with $k_P = 1, k_I = 0.5, k_D = 0.1$). As expected, they yield quite different results. The trajectories obtained using our proposed energy-optimal closest-point projection and the traditional approach that bases the computation on the translational penetration are shown in Figs. 11–12, respectively.

Comparing the trajectories of the center of mass of the body projected on XY-plane for each of these two cases (Fig. 13) one can see that the trajectories diverge at a certain point during the simulation. This point is exactly the instance when the collision is detected and the haptic rendering algorithm is executed.

The rendered forces and torques are also quite different for each of these two cases. The force f_y in the Y-direction and the torque τ_z in the XY-plane are shown in Figs. 14–15,

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

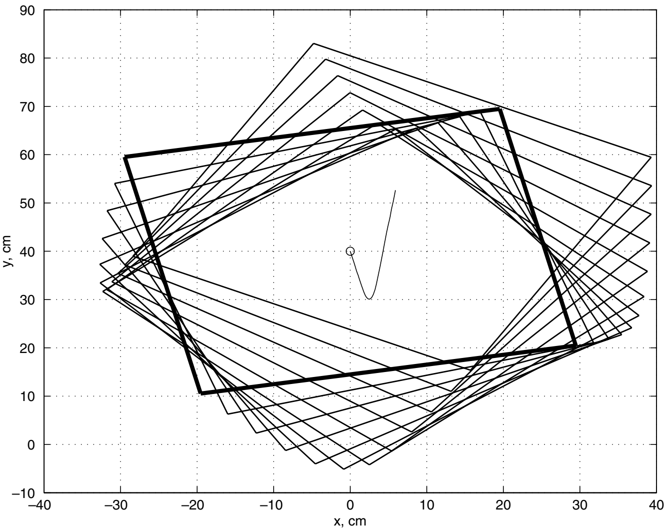

Fig. 12. The trajectory of the rigid body and its center of mass (solid line) projected on the XY-plane using the existing penalty-based method. The initial configuration of the body is shown in bold and the initial position of its center of mass is represented with a circle.

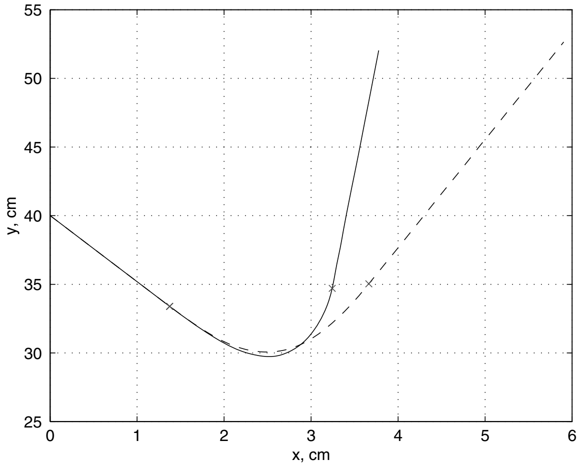

Fig. 13. The trajectory of the center of mass projected on the XY-plane using the proposed energy-optimal method (solid line) and the traditional approach (dashed line). Cross-marks show the starting and ending points of the collision.

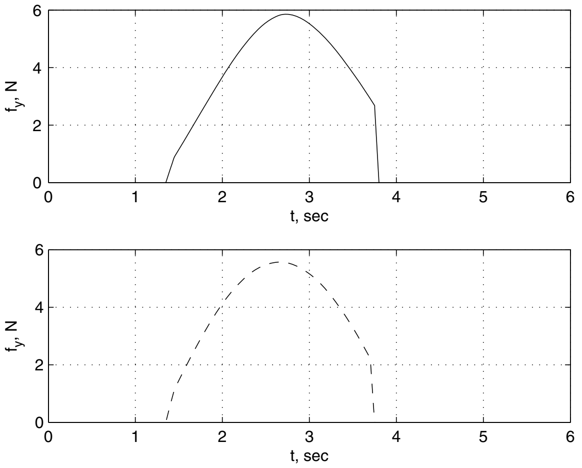

Fig. 14. The rendered force in the Y-direction computed by the proposed energy-optimal method (solid line) and the traditional approach (dashed line).

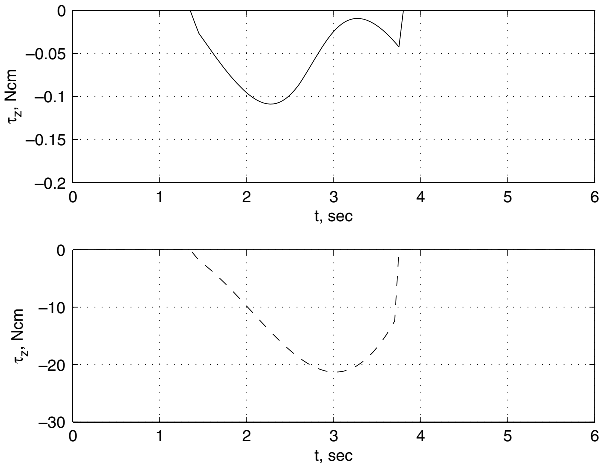

Fig. 15. The rendered torque in the XY-plane using the proposed energy-optimal method (solid line) and the traditional approach (dashed line).

respectively. The figures show that the two approaches produce similar forces, but that the traditional penalty-based approach results in a significantly higher value of the torque, possibly exceeding the admissible torque range of the haptic device.

5. Discussion

Penalty-based haptic rendering techniques rely on penetration depth estimation to compute forces and torques displayed by the haptic device. Existing approaches define penetration depth strictly in the translational sense and might not handle the object rotations properly. This paper proposes a method where the full configuration of the object (translation and rotation) is taken into account when computing the penetration depth. The presented numerical examples show that when the rigid body is rotated into the surface, the contact wrench computed by the proposed method has both torque and force components but differs substantially from the one computed using the traditional methods. This is due to the fact that in the proposed method full configuration is considered, not only its translational part. We hypothesize that the proposed approach could result in a more realistic haptic simulation, but user studies are needed to confirm the hypothesis.

The method can be easily extended to the case of multi-point collision by adding the wrenches computed for each of the collision points. Other issues such as the ability to deal with the pop-through artifacts, force discontinuities, and friction can be dealt with in exactly the same way as in the existing penalty-based approaches. For instance, implicit integration[5] can be used to deal with force discontinuities, and recent work[17] suggests how to take friction into account.

6. Conclusion

The proposed haptic simulation technique provides a theoretical basis for realistic haptic rendering of 6-DOF virtual environments. It is based on the analysis of the configuration matrix of the user-controlled haptic object and determining its energy optimal projection onto the

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press

set of admissible configurations. We have shown that the configuration space allows for more accurate analysis of rigid body motion.

There are two ways to implement our proposed technique: by direct numerical optimization and using an analytical approximation. The main limitation of direct numerical optimization is its computational cost. It can successfully work only on fast computers and in simple cases. As an alternative, we describe an analytical method to approximate the solution. In the planar case, the solution can be obtained in the closed form. By bisecting the object with a one-parameter family of planes, leading in each case to a planar problem, the solution in the general case can be obtained by a one-dimensional search. The proposed analytical approximation is quite fast and its accuracy can be adjusted to take advantage of the available computational power.

While the examples show that the proposed haptic rendering algorithm produces forces and torques that might be more appropriate for haptic simulation, an important step in evaluating haptic algorithms are user evaluations. These are beyond the scope of the current paper and are left for future work. It is also worth mentioning that although the proposed method assumes that a 6-DOF haptic device is used for rendering, it provides a theoretical basis for studying how a general 6-DOF wrench can be rendered with a low-degree-of-freedom haptic device.

Acknowledgments

This research was supported in part by NSF grants IIS-0093581, CCR-0330342, and CMS-0600658. Additional support was provided by the College of Dentistry at the University of Illinois at Chicago.

References

1. K. Salisbury, F. Conti and F. Barbagli, "Haptic rendering: Introductory concepts," *IEEE Comput. Graph. Appl.* **24**(2), 24–32 (2004).
2. M. A. Otaduy and M. C. Lin, *High Fidelity Haptic Rendering*, vol. 1 of *Synthesis Lectures on Computer Graphics*. (Morgan & Claypool Publishers, San Rafael, CA, 2006).
3. M. Moore and J. Wilhelms, "Collision Detection and Response for Computer Animation," *Proceedings of the 15th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 88)*, vol. 22, Atlanta, GA (1988) pp. 289–294.
4. Y. J. Kim, M. A. Otaduy, M. C. Lin and D. Manocha, "Six-degree-of-freedom haptic rendering using localized contact computations," *Presence: Teleop. Virt.* **12**(3), 277–295 (2003).
5. M. A. Otaduy and M. C. Lin, "Stable and Responsive Six-Degree-of-Freedom Haptic Manipulation using Implicit Integration," *Proceedings of the World Haptics Conference*, Pisa, Italy (2005) pp. 247–256.
6. J. E. Colgate, M. C. Stanley and J. M. Brown, "Issues in the Haptic Display of Tool Use," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL (1994) pp. 295–300.
7. C. B. Zilles and J. K. Salisbury, "A Constraint-Based God-Object Method for Haptic Display," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA (1995) pp. 146–151.
8. D. C. Ruspini, K. Kolarov and O. Khatib, "The Haptic Display of Complex Graphical Environments," *Proceedings of the 24th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 97)*, Los Angeles, CA (1997) pp. 345–352.
9. M. Ortega, S. Redon and S. Coquillart, "A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies," *Proceedings of the IEEE Virtual Reality*, Alexandria, VA (2006) pp. 197–204.
10. D. Constantinescu, S. E. Sacludean and E. A. Croft, "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Trans. Robot.* **21**(3), 309–323 (2005).
11. W. A. McNeely, K. D. Puterbaugh and J. J. Troy, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," *Proceedings of the 26th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, New York, NY, USA (1999) pp. 401–408.
12. M. A. Otaduy, N. Jain, A. Sud and M. C. Lin, "Haptic Display of Interaction Between Textured Models," *Proceedings of IEEE Visualization Conference*, Austin, TX (2004) pp. 297–304.
13. L. Zhang, G. Varadhan, Y. J. Kim and D. Manocha, "Generalized Penetration Depth Computation," *Proceedings of ACM Solid and Physical Modeling Conference (SPM06)*, Wales, UK (2006) pp. 173–184.
14. L. Zhang, Y. J. Kim and D. Manocha, "C-Dist: Efficient Distance Computation for Rigid and Articulated Models in Configuration Space," *Proceedings of ACM Solid and Physical Modeling Conference (SPM07)*, Beijing, China (2007) pp. 159–169.
15. L. Zhang, Y. J. Kim and D. Manocha, "A Fast and Practical Algorithm for Generalized Penetration Depth Computation," *Proceedings of Robotics: Science and Systems (RSS07)*, Atlanta, GA (2007).
16. C. Kane, E. A. Repetto, M. Ortiz and J. E. Marsden, "Finite element analysis of nonsmooth contact," *Comput. Methods Appl. Mech. Engng.* **180**(1–2), 1–26 (1999).
17. Q. Luo and J. Xiao, "Physically accurate haptic rendering with dynamic effects," *IEEE Comput. Graph. and Appl.* **24**(6), 60–69 (2004).
18. R. M. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (Boca Raton, FL, CRC Press, 1994).
19. M. Žefran and F. Bullo, "Lagrangian Dynamics," *Robotics and Automation Handbook* (T. R. Kurfess, ed.), (CRC Press, 2005) pp. 5-1–5-16.
20. W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, 2nd ed. (Academic Press, New York, 1986).
21. Q. Lin and J. W. Burdick, "Objective and frame-invariant kinematic metric functions for rigid bodies," *Intl J. Robot. Res.* **19**(6), 612–625 (2000).
22. M. Žefran and V. Kumar, "A geometrical approach to the study of the Cartesian stiffness matrix," *J. Mech. Design* **124**(1), 30–38 (2002).
23. M. C. Lin and J. F. Canny, "A Fast Algorithm for Incremental Distance Calculation," *IEEE International Conference on Robotics and Automation*, Sacramento, CA (1991) pp. 1008–1014.
24. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (Dover, New York, 1972).
25. S. Booth, F. De Angelis and T. Schmidt-Tjarksen, "The Influence of Changing Haptic Refresh-Rate on Subjective User Experiences – Lessons for Effective Touch-Based Applications," *EuroHaptics 2003 Conference Proceedings*, Dublin, Ireland (2003) pp. 374–383.
26. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1980).

https://doi.org/10.1017/S0263574708004207 Published online by Cambridge University Press