A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties

Michael Ortega, Stephane Redon, Sabine Coquillart

A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties

Michal Ortega [⋆] , Stephane Redon [†] , Sabine Coquillart [‡]

Abstract— This paper describes a generalization of the god-object method for haptic interaction between rigid bodies. Our approach separates the computation of the motion of the six degree-of-freedom god-object from the computation of the force applied to the user. The motion of the god-object is computed using continuous collision detection and constraint-based quasi-statics, which enables high-quality haptic interaction between contacting rigid bodies. The force applied to the user is computed using a novel constraint-based quasi-static approach, which allows us to suppress force artifacts typically found in previous methods. The constraint-based force applied to the user, which handles any number of simultaneous contact points, is computed within a few microseconds, while the update of the configuration of the rigid god-object is performed within a few milliseconds for rigid bodies containing up to tens of thousands of triangles. Our approach has been successfully tested on complex benchmarks. Our results show that the separation into asynchronous processes allows us to satisfy the different update rates required by the haptic and the visual displays. Force shading and textures can be added and enlarge the range of haptic perception of a virtual environment.

This paper is an extension of [1].

Index Terms— Haptics, God-Object, Six degrees of freedom, Rigid bodies, Constraint-based quasi-statics

I. INTRODUCTION

HAPTIC display of rigid bodies has the potential to improve the interaction between a human and a virtual environment by providing the user with the ability to touch and feel the geometric details of the virtual objects. Typical applications include CAD/CAM design, virtual prototyping, scientific visualization and medical simulation.

Because of the high computational requirements of haptic rendering, however, finding effective methods is still a great challenge. A classical three degree-of-freedom method for haptic display of the interaction of a point and a virtual object was introduced by Zilles and Salisbury [2]. The two main benefits of their approach are (a) a non-penetrating simulation of the motion of the point as it slides on the surface of the obstacles; (b) a constraint-based computation of the force

[⋆] **Michal Ortega** *is with PSA Peugeot Citroën and i3D-INRIA Rhône-Alpes-GRAVIR , 655 avenue de l'Europe 38330 Montbonnot St Martin, France*
e-mail: michael.ortega@inria.fr

[†] **Stephane Redon** *is with i3D-INRIA Rhône-Alpes - GRAVIR , 655 avenue de l'Europe 38330 Montbonnot St Martin, France*
e-mail: stephane.redon@inria.fr

[‡] **Sabine Coquillart** *is with i3D-INRIA Rhône-Alpes - GRAVIR , 655 avenue de l'Europe 38330 Montbonnot St Martin, France*
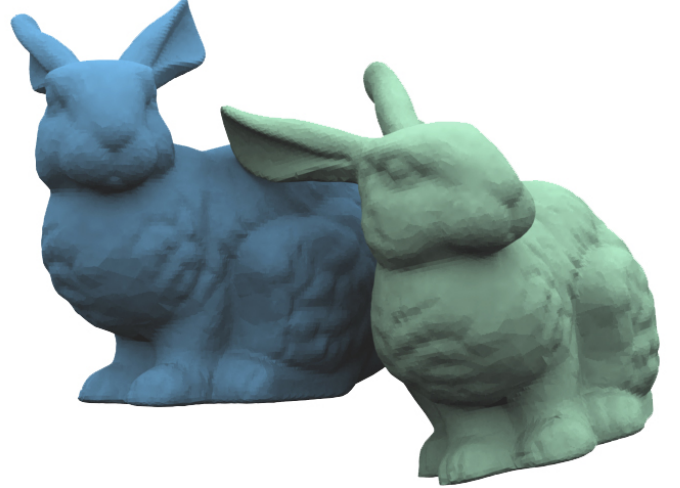*e-mail: sabine.coquillart@inria.fr*

Fig. 1. **Haptic interaction with Stanford Bunnies.** The approach described in this paper allows us to provide a user with high-quality haptic display of contacting rigid bodies (here, two Stanford bunnies containing about 27,000 triangles each). Our constraint-based force computation method allows the manipulated object to come in contact with and slide on the environment obstacles without penetrating them, while providing the user with precise haptic display, where each vertex, edge and face can potentially be felt.

applied to the user, which results in a force orthogonal to the constraints. These features are highly desirable, in that non-interpenetration of virtual objects is known to increase their perceived stiffness [3], and that an incorrect orientation of the force has been shown to perturb the perceived orientation of the virtual surfaces [4].

Although several six degree-of-freedom haptic rendering methods have been proposed (see Section II), none seems to preserve all of the properties of the initial three degree-of-freedom approach introduced by Zilles and Salisbury [2]: these methods might allow the virtual objects to interpenetrate, or they use some form of virtual coupling [5] which can lead to disturbing force artifacts, by modifying the orientation of the force applied to the user. In this paper, we propose what seems to be the first six degree-of-freedom constraint-based method that prevents both these visual and haptic artifacts. Especially, we make the following contributions:

- **Six degree-of-freedom god-object method:** we extend the three degree-of-freedom god-object method proposed by Zilles and Salisbury [2] to six degree-of-freedom haptic interaction between rigid bodies.
- **High-quality god-object simulation:** our god-object simulation method prevents any interpenetration between

the virtual objects, while allowing the god-object to precisely contact and slide on the surface of the obstacles. This results in highly detailed haptic rendering of the objects geometries, and increases the perceived stiffness of the virtual objects [3].

- **Constraint-based force computation:** we introduce a novel constraint-based quasi-static approach to compute the motion of the god-object and the force applied to the user. The constraint-based approach is physically-based, handles any number of simultaneous contact points, and yields constraint forces that are orthogonal to the constraints, thereby rendering correct surface orientations to the user. Furthermore, we show that our constraint-based quasi-static approach can only dissipate the energy transmitted to the god-object. This helps us improve the stability of the haptic display.

The paper is organized as follows. Section II provides a summary of related work. Section III gives an overview of our approach. Section IV describes how we compute the motion of the god-object to ensure realistic haptic interaction with rigid bodies. Section V presents our novel constraint-based quasi-static approach to computing the force applied to the user. Section VI discusses methods for producing haptic effects for surface perception such as force shading and textures. Section VII demonstrates our approach on several benchmarks and shows how our approach is able to provide the user with high-quality haptic display of contacting rigid bodies. We also discuss the benefits and limitations of our approach. Finally, Section VIII concludes and details several future research directions.

II. RELATED WORK

Haptic display of virtual objects has been an active area of research over the last decade. In 1995, Zilles and Salisbury [2] proposed what appears to be the first constraint-based method for three degree-of-freedom haptic rendering of generic polygonal objects. They introduced the *god-object*, an idealized representation of the position of the haptic device that is constrained to the surface of the obstacles. In their three degree-of-freedom approach, the location of the god-object minimizes at each time step the distance to the haptic device; the difference between the two positions provides the force direction. Ruspini *et al.* [6] extends this approach by replacing the god-object by a small sphere, and proposes methods to smooth the object surface and add friction. Niemayer and Mitra [7] proposes dynamic proxies to better simulate dynamic effects. Several authors have proposed to extend the virtual proxy approach to three degree-of-freedom interaction with objects defined by implicit representations [8], [9].

Some authors have proposed six degree-of-freedom haptic display algorithms. McNeely *et al.* [10] proposed a voxel sampling method. Johnson and Willemsen [11] use local minimum distances to compute the force applied to the user. Gregory *et al.* [12] extend the virtual proxy approach to six degrees of freedom and estimate the local penetration depth to compute the force and torque applied to the user. These methods, like most six degree-of-freedom haptic display

methods [13], [14], [15], [16], [17], [18], do not attempt to prevent the interpenetration between the virtual objects, which might lead to miss some collisions between the virtual objects and can lead to the well-known *pop-through effect*, where the virtual proxy can traverse thin objects or objects parts [6], thereby degrading the perception of geometric details. Berkelman *et al.* [19] have proposed a general constraint-based method for six degree-of-freedom interaction with rigid bodies. However, their approach includes a virtual coupling [5] which leads to perceptible force artifacts (see discussion in Section VII). Recent work on stable six degree-of-freedom interaction by Otaduy and Lin [20], however, has shown that the force artifacts created by a virtual coupling can be reduced through the use of an implicit integration method.

To the best of our knowledge, the approach described in this paper seems to be the first six degree-of-freedom constraint-based haptic rendering method that does not suffer from the visual or haptic artifacts of previous approaches (*i.e.* interpenetrations, forces felt at a distance, or artificial friction and sticking).
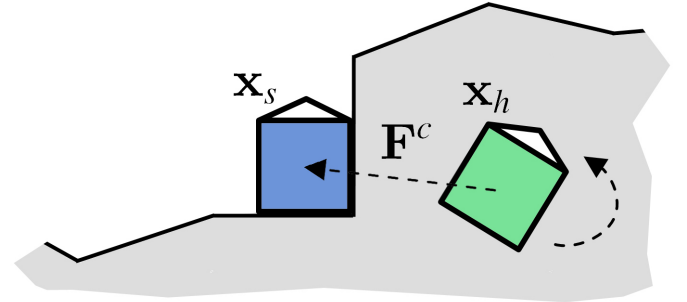
III. OVERVIEW

Fig. 2. **Six degree-of-freedom god-object.** Although the haptic device penetrates the environment obstacles (configuration $\mathbf{x}_h$), the god-object is constrained to remain on the surface of the obstacles (configuration $\mathbf{x}_s$). We propose new algorithms to compute the motion of the god-object and the force applied to the user based on the discrepancy between these two configurations.

Our method extends the classical three degree-of-freedom constraint-based method by Zilles and Salisbury [2] by employing a *six degree-of-freedom god-object*, *i.e.* an idealized representation of the haptic device that is constrained to remain on the surface of the environment obstacles when the haptic device penetrates the environment obstacles (see Figure 2). At each time step, we attempt to reduce the discrepancy between two rigid reference frames: one attached to the haptic device, and one attached to the virtual object. We typically place the origin at the center of gravity of the virtual object, although any point can be chosen. Only the god-object is displayed (and not the actual configuration of the haptic device), so that even when the haptic device penetrates the environment obstacles, the user only sees the rigid body that he manipulates in a realistic, contacting only configuration. As a result, the user feels that the rigid body he or she is manipulating is correctly sliding on the surface of the obstacles. The motion of the god-object and the force applied to the user are computed from the discrepancy between the configurations of the god-object and the haptic device, thanks to a novel constraint-based quasi-
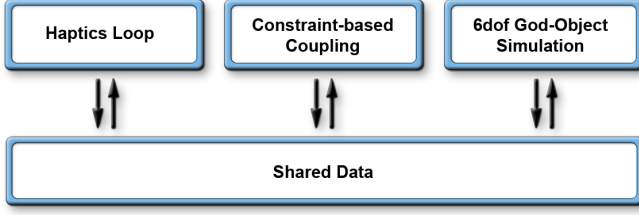
Fig. 3. **Schematic representation of our method.** Our method for haptic display of six degree-of-freedom manipulation of rigid bodies is divided in three asynchronous blocks (see Section III).

static approach which suppresses visual and haptic artifacts typically found in previous approaches.

Our algorithm is divided in three asynchronous loops: (a) the god-object simulation loop, which updates the configuration of the god-object based on the configuration of the haptic device and the environment obstacles; (b) the constraint-based coupling loop, which determines the constraint-based force applied to the user based on the configurations of the god-object and the haptic device, as well as the current set of contact points and normals; (c) the haptics loop, which controls an impedance-like haptic device which reads the force that has to be applied to the user and writes the current configuration of the haptic device (see Figure 3). The haptics loop is considered as a generic black box, and this paper focuses on the two other processes, *i.e.* the god-object simulation loop and the constraint-based coupling loop.

IV. SIX DEGREE-OF-FREEDOM GOD-OBJECT SIMULATION
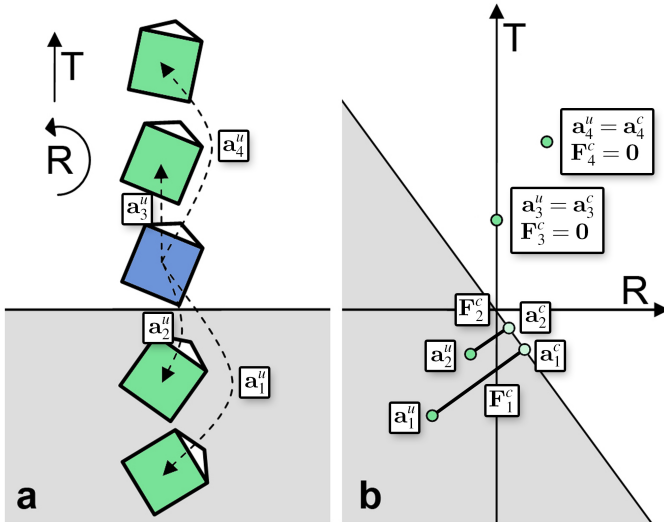
A. Overview

Fig. 4. **Constraint-based force computation.** Our method uses Gauss' least constraints principle to compute the constrained motion of the god-object and the constraint-based force applied to the user (see Sections IV and V).

The motion of the god-object is computed based on the relative configurations of the haptic device and the god-object, as well as the current set of contact points. Precisely, we perform a constraint-based quasi-static simulation of the god-object according to the following *god-object simulation algorithm*:

1) **Data retrieval:** The six-dimensional configuration $\mathbf{x}_h$ of the haptic device is retrieved from the shared data (see Figure 3).
2) **Unconstrained acceleration computation:** The unconstrained six-dimensional acceleration $\mathbf{a}^u$ of the god-object is computed from $\mathbf{x}_h$ and the six-dimensional configuration $\mathbf{x}_s$ of the god-object:

$$\mathbf{a}^u = k_s(\mathbf{x}_h - \mathbf{x}_s),$$

where k_s is a coupling constant ($k_s = 0.5$ in our implementation). This is similar to the virtual coupling method [5], except that we directly control the acceleration of the god-object. Because the motion of the god-object is quasi-static, this amounts to directly control the displacement of the god-object.
3) **Constraint-based quasi-static computations:** The constrained acceleration $\mathbf{a}^c$ of the god-object is computed based on the current contact information (*i.e.* the one resulting from the previous god-object simulation step) and the unconstrained acceleration $\mathbf{a}^u$. This involves forming the 6×6 god-object mass matrix $\mathbf{M}$ and the $6 \times m$ contact Jacobian $\mathbf{J}$, where m is the number of contact points (see details below).
4) **Collision detection:** The *target configuration* of the god-object is computed from its constrained acceleration using an explicit Euler integration step. We use the continuous collision detection algorithm introduced by Redon *et al.* [21] to detect collisions on a path interpolating the current and target god-object configurations. If the interpolating path is free of collisions, the god-object is placed in the target configuration. If a new contact occurs, however, the continuous collision detection algorithm determines the first contacting configuration along the interpolating path, as well as the new contact positions and normals. The configuration reached by the god-object at the end of this step is the *new god-object configuration*.
5) **Constraints transmission:** The matrices $\mathbf{M}$ and $\mathbf{J}$ corresponding to the new god-object configuration are written to the shared data, so that they can be retrieved by the constraint-based coupling loop to compute the constraint-based force applied to the user.

The god-object simulation loop ensures that the god-object attempts to reach the same configuration (position and orientation) as the haptic device. Continuous collision detection and constraint-based quasi-statics allow the god-object to slide on virtual obstacles without penetrating them as it tries to reach the haptic device. In the following, we describe how we derive the constraint-based quasi-statics of the god-object using Gauss' least constraint principle.

B. Constraint-based God-Object Quasi-Statics

Let $\mathbf{a} = (\mathbf{a}_G, \alpha)^T$ denote the generalized (six-dimensional) acceleration of the god-object, where $\mathbf{a}_G$ and α are respectively the linear acceleration and the angular acceleration of the god-object. The set of *possible* accelerations is easily determined from the contact positions and normals provided by the continuous collision detection algorithms. Let I_k and
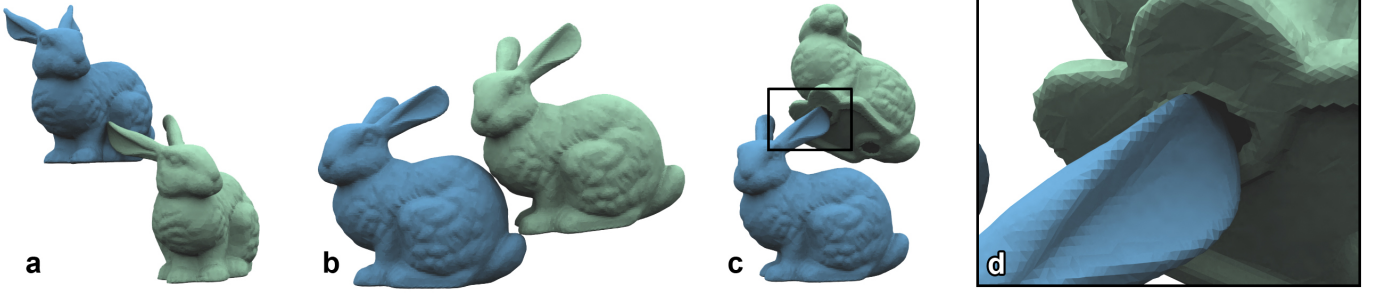
Fig. 5. **Haptic interaction with Stanford Bunnies.** The user manipulates the green bunny. **a:** the ear of the green bunny slides in a ridge of the blue bunny. **b:** continuous collision detection and constraint-based quasi-statics allows the manipulated object to precisely contact and slide on the obstacles. **c-d:** our method provides the user with the ability to precisely feel the contact between pairs of triangles, resulting in highly detailed haptic display of contacting rigid bodies.

$\mathbf{n}_k$ respectively denote the position and normal of the k-th contact point, $1 \leqslant k \leqslant m$. Assuming the normal $\mathbf{n}_k$ is directed towards the exterior of the environment obstacle, the acceleration of the god-object must satisfy the following *non-penetration constraint* [22]: $\mathbf{a}_G^T \mathbf{n}_k + \boldsymbol{\alpha}^T (GI_k \times \mathbf{n}_k) \geqslant 0$, where GI_k is the vector from the center of inertia G of the god-object to the contact point I_k. Note the absence of velocity-dependent term in the non-penetration constraint, as the quasi-static assumption implies that the velocity of the god-object is zero at all times. These m non-penetration constraints can be concatenated to form a single constraint on the generalized acceleration of the god-object: $\mathbf{J}\mathbf{a} \geqslant \mathbf{0}$, where $\mathbf{J}$ is a $m \times 6$ *Jacobian*.

Gauss' principle states that the constrained generalized acceleration $\mathbf{a}^c = (\mathbf{a}_G^c, \boldsymbol{\alpha}^c)^T$ of the god-object minimizes the following function [23]:

$$\mathscr{G}(\mathbf{a}) = \frac{1}{2}(\mathbf{a} - \mathbf{a}^u)^T \mathbf{M}(\mathbf{a} - \mathbf{a}^u) = \frac{1}{2}||\mathbf{a} - \mathbf{a}^u||_{\mathbf{M}}^2, \quad (1)$$

that is, the *kinetic distance* $||\mathbf{a}^c - \mathbf{a}^u||_{\mathbf{M}}$ between the constrained acceleration $\mathbf{a}^c$ and the unconstrained acceleration $\mathbf{a}^u$, over the set of possible accelerations $\{\mathbf{a} : \mathbf{J}\mathbf{a} \geqslant \mathbf{0}\}$. In other words, the constrained acceleration $\mathbf{a}^c$ is the (non-euclidean) projection of the unconstrained acceleration $\mathbf{a}^u$ onto the set of possible accelerations. This projection problem is solved using Wilhelmsen's projection algorithm [24]. Note that the matrices $\mathbf{M}$ and $\mathbf{J}$ contain all the necessary and sufficient information to compute the constrained motion of the god-object.

V. CONSTRAINT-BASED FORCE COMPUTATION

The constraint-based coupling loop determines the forces applied to the user based on the configuration of the haptic device and the contact information sent by the god-object simulation loop. Essentially, the constraint-based coupling loop performs the same constraint-based quasi-static computations as in the god-object simulation loop, but *assuming the configuration of the god-object is fixed*. This suppresses the need for collision detection in the constraint-based coupling loop, and allows us to compute the constraint-based force applied to the user within a few *microseconds* (see Section VII). Precisely, the constraint-based force applied to the user is computed according to the following *constraint-based force computation algorithm*:

1) **Data retrieval:** The configuration $\mathbf{x}_h$ of the haptic device and the configuration $\mathbf{x}_s$ of the god-object are read from the shared data, as well as the matrices $\mathbf{M}$ and $\mathbf{J}$, computed in the god-object simulation loop, which describe the local quasi-statics of the god-object.

2) **Unconstrained acceleration computation:** As in the god-object simulation loop, the unconstrained six-dimensional acceleration $\mathbf{a}^u$ of the god-object is computed from $\mathbf{x}_h$ and the six-dimensional configuration $\mathbf{x}_s$ of the god-object ($\mathbf{a}^u = k_s(\mathbf{x}_h - \mathbf{x}_s)$).

3) **Constraint-based force computation:** The constrained acceleration $\mathbf{a}^c$ of the god-object is computed from the unconstrained acceleration $\mathbf{a}^u$ and the matrices $\mathbf{M}$ and $\mathbf{J}$ retrieved from the shared data, by solving Gauss' projection problem. The constraint-based force to be applied to the user is then $\mathbf{F}^c = k_h \mathbf{M}(\mathbf{a}^c - \mathbf{a}^u)$, where k_h is a coupling constant[1].

4) **Force transmission:** The constraint-based force $\mathbf{F}^c$ is written to the shared data. It will be read by the haptic loop, for application to the user.

Figure 4 demonstrates this algorithm in the case of a god-object in contact with an obstacle. For clarity, only two degrees of freedom are allowed: a vertical translation and a rotation whose axis is orthogonal to the plane of the figure. Figure 4.a shows the god-object contacting the obstacle (in blue), and four successive configurations of the haptic device (in green), as well as the resulting unconstrained accelerations $\mathbf{a}_1^u$, ..., $\mathbf{a}_4^u$. Figure 4.b shows the corresponding two-dimensional *motion-space*, *i.e.* the space of accelerations, and the linearized non-penetration constraint resulting from the contact point (the diagonal line). The possible accelerations are above this diagonal line. Projecting the unconstrained accelerations $\mathbf{a}_1^u$, ..., $\mathbf{a}_4^u$ on the set of possible accelerations yields the constrained accelerations $\mathbf{a}_1^c$, ..., $\mathbf{a}_4^c$, as well as the corresponding constraint forces $\mathbf{F}_1^c$, ..., $\mathbf{F}_4^c$ applied to the user. Haptic configurations 1 and 2 result in a force and a torque which attempt to bring the haptic device back to a configuration reachable by the god-object, while haptic configurations 3 and 4, which correspond to accelerations satisfying the non-penetration constraint, do not generate any force.

Note that, because the configuration $\mathbf{x}_s$ of the god-object is not updated in the constraint-based coupling loop, the matrices

---

[1]Different constants can be used for the translational and rotational parts, but this might lead to constraint forces that are not orthogonal to the non-penetration constraints (see Section VII).
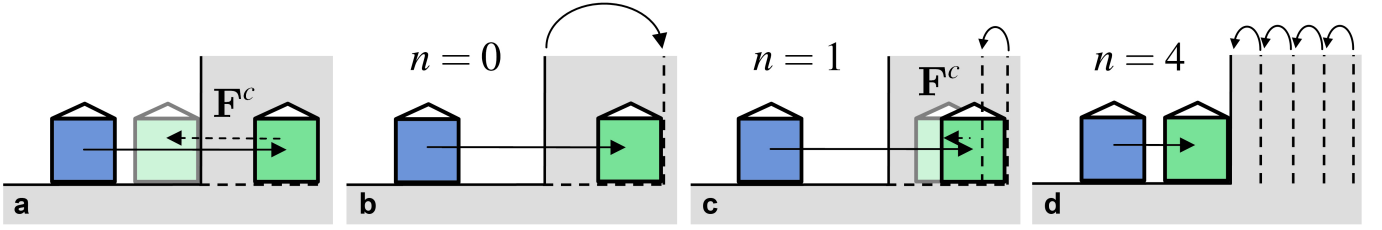
Fig. 6. **Constraints adaptation.** When a new constraint (here a vertical plane) appears that would create too large a constraint force, it is first translated so that the constraint is satisfied by the current haptic device configuration, then progressively returned to its initial position. This helps us smooth the force felt by the user, while ensuring that small discontinuities signaling new contact points are felt.

$\mathbf{M}$ and $\mathbf{J}$ do not have to be updated either[2]. Hence, only the configuration of the haptic device changes, and the main computation involved is the determination of the constrained acceleration $\mathbf{a}^c$, which can be performed very efficiently (see Section VII).

When a new set of constraints is available, some of the new non-penetration constraints might not be satisfied by the current configuration of the haptic device (see Figure 6.a). This might create a large constraint force if the user has largely penetrated those new constraints. In order to smooth the constraint-based force applied to the user and reduce potentially large forces created by delays in the update of the set of constraints, we generalize the method introduced by Mark *et al.* [25]. Assume a new constraint $\mathbf{J}_k\mathbf{a} \geqslant 0$ on the acceleration $\mathbf{a}$ of the god-object occurs, where $\mathbf{J}_k$ is a six-dimensional row vector (a row of the Jacobian). Assume that this constraint is not satisfied at time 0, when the new set of constraints becomes available, *i.e.* that the configuration of the haptic device is such that $\mathbf{J}_k\mathbf{a}^u = d_k < 0$. We initially *offset* this constraint: the constraint becomes $\mathbf{J}_k\mathbf{a}^u \geqslant f_k(t)$, where f_k is a monotonously increasing time-dependent function such that $f_k(0) = d_k$ and $f_k(\Delta t) = 0$. This constraint is thus satisfied when the set of constraints is progressively turns into the constraint that should be enforced (*i.e.* after a time Δt, see Figure 6.b-d). In order 6.b-d). In order to provide the user with a slight force discontinuity and improve the perception of new constraints, however, we perform this interpolation only if $d_k \leqslant \varepsilon$, where ε acts as a user-defined discontinuity threshold ($\varepsilon < 0$). We leave the formal evaluation of the influence of ε on the haptic perception of new contacts and on the overall stability of the algorithm for future work.

The combination of the god-object simulation loop and the constraint-based coupling loop results in the perception of six degree-of-freedom constraint forces as the user manipulates the virtual object and slides on the virtual obstacles.

VI. HAPTIC SURFACE PROPERTIES

The six degree-of-freedom constraint-based method proposed here provides a force orthogonal to the non-penetration constraints. No force artifacts are felt by the user, such as artificial friction or sticking effect. The force vector direction can now be controlled and perturbed for providing haptic surface properties like force shading or texture. The two

[2]In our implementation, a flag is used to signal the arrival of a new set of constraints to the constraint-based coupling loop. This flag, written to the shared data by the god-object simulation loop, allows us to avoid re-reading the matrices $\mathbf{M}$, $\mathbf{J}$, and the god-object configuration $\mathbf{x}_s$, which further speeds up the constraint-based coupling loop.

following sections demonstrate how such effects can be added, by modifying either the constraints or the force applied to the user.

A. Smooth Surfaces

Our current implementation uses a continuous collision detection method suitable for polygonal objects. As a result, smooth shaped objects approximated by polygonal meshes feel like polyhedral surfaces, due to the discontinuity at the polygon edges. To avoid that, Morgenbesser and Srinivasan [26] have been the first to adapt the well known Phong method [27] for smoothing polygonal meshes. They demonstrated that a similar haptic effect, called force shading, can give the illusion of a haptically smooth shape.

More recently, Ruspini *et al.* [6] also proposed to adapt the graphical methods using the virtual proxy approach. Compared to the Morgenbesser approach, their force shading method allows them to handle situations involving multiple intersections between the proxy and shaded surfaces at the same time.

Like the Ruspini *et al.* approach, the constraint-based method proposed in this paper allows us to adapt the Phong method [27]. At each point on a mesh polygon, a new vector is computed by interpolating the normals from the vertices of the polygon. This new normal is used to compute the illumination of the model at this point. Consequently, the edges of the polygonal mesh do not appear, and the shape appears to be smooth. The same idea is used for force shading.

The following sections explain the link between the force vector direction and the surface normal, followed by the description of the force shading algorithm. Finally, they show how force shading can be efficiently computed in our asynchronous algorithm.

1) Surface normal and force rendered: as described in Section IV and V, the computation of the force directly results from the computation of the constrained acceleration, which itself uses both the unconstrained acceleration and the contact information (or constraint space). The latter is mainly defined by the surface normal for each contact point between the god-object and the shape. Consequently, changing the surface normals in the contact information will change the direction of the force vector.

2) Basic algorithm: using contact positions, similarly to the Phong approach, the algorithm proceeds by first computing the interpolated contact normals at each position of the contact points. These vectors are used to create a new constraint space,
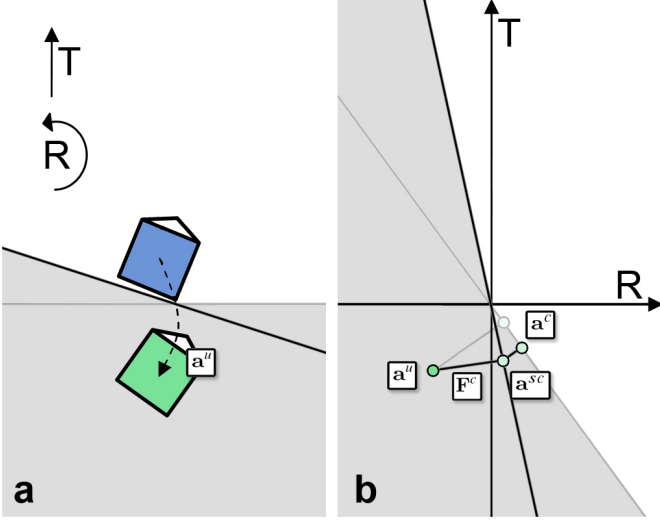
Fig. 7. **Two computation passes.** The Gauss' least constraints principal is used twice to compute the force shading constrained acceleration a^{sc} and the final constrained acceleration a^c. The first pass uses the force shading constraint space, while the second pass computes the motion of the god-object with the original constraint space and the force shading constrained acceleration as an unconstrained acceleration.

Fig. 8. **Smoothing Effect. a:** The edge is felt because of the discontinuous force implied by the change in the normal direction. **b:** Thanks to the use of the vertex normal **N**, the force shading method avoids the discontinuity, and smoothes the edge.

called force shading constraint space. The rest of the algorithm consists of two computation passes (*cf* Figure 7), *i.e.* the computation of the new direction of the force vector and the computation of the new god-object configuration.

- **Force vector direction.** First, a force shading constrained acceleration is computed from the unconstrained acceleration and the force shading constraint space. Next, the computation of the force is done with this new acceleration and the original unconstrained acceleration. At this point, and without the next stage, the force rendered by the haptic device will give the illusion of a non-flat mesh polygon, but the edges are still felt. Next stage explains how to avoid that.
- **Constrained acceleration.** As seen in Figure 8.a, with the six degree-of-freedom god-object method, a discontinuity occurs when the user reaches an edge of the shape. Indeed, such an effect is provided by the computation of the constraint acceleration, which is always as close as possible to the unconstrained acceleration. Even with the computation of the perturbed force direction described in the stage before, this sudden change in the configuration of the god-object makes the user feel the edges of the polygonal mesh. To avoid that, the force shading constraint acceleration is used as an unconstrained acceleration and combined with the original constraint space to compute a final acceleration for the god-object (*cf* Figure 7). Figure 8.b shows the successive god-object configurations when such an approach is used.

3) Optimization with the asynchronous process: the computation described above is a time-consuming computation, because of the double constraint-based quasi-static computation. This can be optimized by exploiting the asynchronous aspect
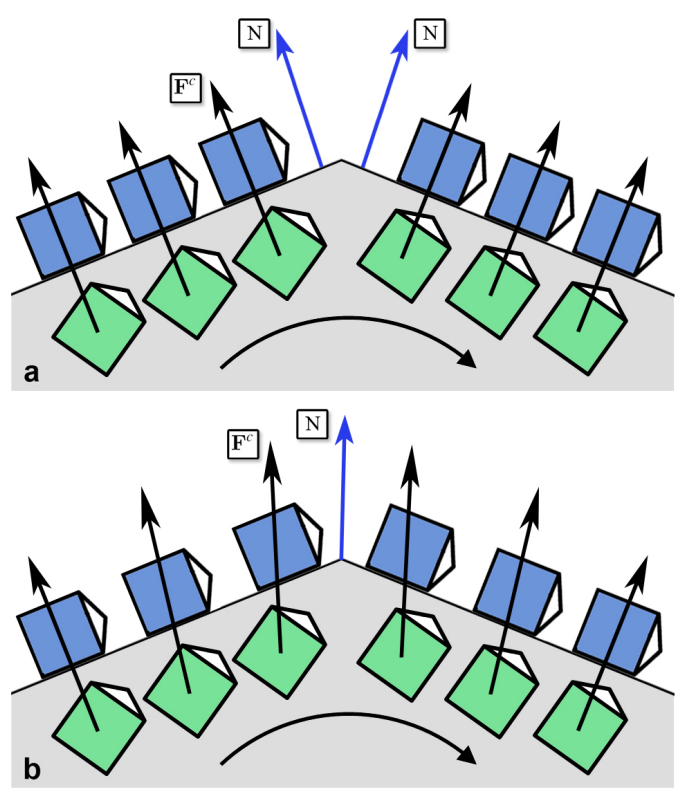
of the proposed algorithm, and by implementing one step in each process (*i.e.* the simulation and the coupling loops).

The force shading constraint space is created by the simulation loop, and written to the shared data. In parallel, the coupling loop uses the last force shaded constraint space retrieved and computes the force shading constrained acceleration which is also written in the shared data. Consequently, instead of the unconstrained acceleration, the force shading constrained acceleration is computed by the simulation loop using the original constraint space to create the new constraint acceleration of the god-object.

B. Textures

Except for some recent methods for six degree-of-freedom haptical texture rendering [28], [29], which are not usable here, most of the existing approaches proposed to explore textured surfaces in three degrees of freedom [30], [31], [32]. Minsky [33] were the first to introduce a system to synthesize high-frequency textures for a haptic device. Only in 2D, they used a texture-map method. This approach is an adaptation of the bump-mapping graphical method proposed by Blinn [34]. The approach combines the haptic device location and the map to provide a surface property and a force feedback. This produces a convincing effect of high-frequency textures.

A similar effect can be produced by perturbing the force computed by the six degree-of-freedom constraint-based god-object approach, using a discrete or continuous function at

the contact point position. For example, a sine function along one axis could be sufficient for providing bumps and holes along this axis (*cf* Figure 9). In the case of multiple contact points, the perturbation vector used to modify the force vector direction is defined by averaging the perturbation vector at each contact points.

This method provides high-frequency textures and can be mixed with the force shading effect described above. However, similar to the Minsky *et al.* approach, if the speed of the god-object is too high, or the update rate of the simulation loop is too low, the contact point positions can pass from a hole directly to another one without feeling the bump in between. This implies a limitation in the texture frequency according to the exploration speed and the update rate of the simulation loop.
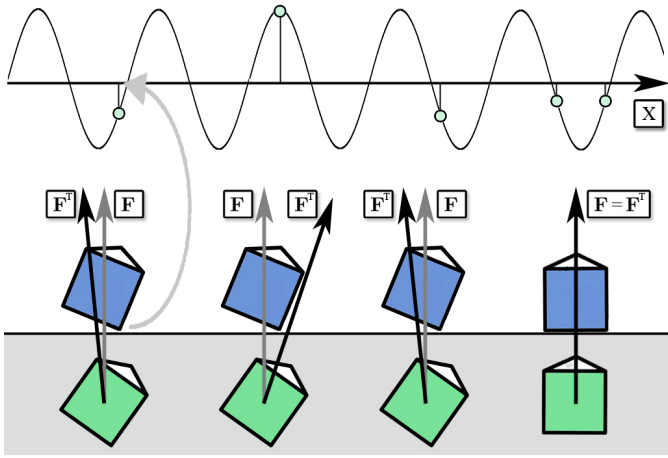

Fig. 9. **Bump and Hole Texture.** The direction of the force vector **F** is perturbed by a sine function. The **x** position of the contact point is an entry of the sine function to find a value for perturbing the direction of the force. The perturbed force $\mathbf{F}^T$ is transmitted to the haptic device, allowing the user to feel the bumps and holes defined by this function.

VII. RESULTS AND DISCUSSION

The validation of our approach is performed on a Stringed Haptic Workbench in which the SPIDAR-G, a tension-based six degree-of-freedom force-feedback device [35], allows a user to interact intuitively on a large two-screen display [36]. The entire algorithm is executed on a 3.2 GHz dual-processor Xeon PC, to which the haptic device is connected. This PC communicates with a cluster of PCs only dedicated to the stereo display on both screens of the Stringed Haptic Workbench. The communication between the Xeon PC and the cluster of PCs is ensured by UDP protocols.

Each of the three main loops is launched in its separate thread. The haptic device thread frequency is fixed by the device: the constraint-based force computed by the constraint-based coupling loop is read from the shared data and applied to the user at 1000 Hz. The frequencies of the constraint-based coupling thread and the god-object simulation thread vary over time, depending on the complexity of the models and the task being performed (see below).
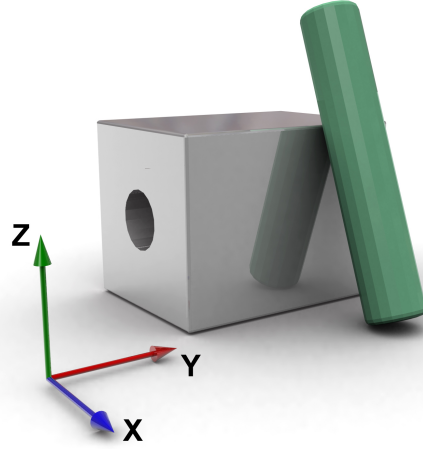

Fig. 10. **The models used in the peg-in-a-hole benchmark.** The peg contains 288 triangles, while the hole contains 280 triangles. The hole is aligned with the Y axis.

A. Peg-in-a-hole Benchmark

We first evaluate the quality and the stability of the haptic interaction in a simple but classical case: the peg-in-a-hole benchmark (see Figure 10). This benchmark is well-known because, although it involves only very simple geometry (here, 288 triangles for the peg and 280 triangles for the box), it has typically been a challenge to provide a stable and realistic haptic display of the insertion of the peg, due to the multiple and potentially redundant contact points occurring during the task [12].

Figure 11 reports several timings and statistics measured during a typical interaction. The first row reports several key configurations tested during the interaction, including (a) sliding the tip of the peg on the top side of the box, (b) laying the peg on the top side of the box and sliding it on the box, (c) pushing on the left side of the box, (d) exploring the right extremity of the hole and (e) inserting the peg in the hole. The second row reports the time required to compute the constraint-based force (see Section V) during the interaction. It can be seen that the constraint-based force is computed in less than 25 *microseconds* throughout the manipulation. The third row shows that the time required to update the configuration of the god-object is always smaller than 10 milliseconds, which is sufficient to prevent any visual lag throughout the manipulation. The fourth row reports the number of simultaneous contact points during the interaction, which can be seen to be fairly limited throughout the manipulation. This can be easily explained by the fact that (a) new contact points rarely occur exactly simultaneously, and (b), compared to other approaches using the interpenetration between virtual objects, constraint-based quasi-static computations tend to limit the apparition of new contact points, since at most twelve of them can be independent (each constraint removes half a degree of freedom). This greatly contributes to the efficiency of the constraint-based coupling loop. Finally, the fifth and the sixth rows reports the Y and Z components of the constraint-based force applied to the user during the interaction. As expected, the Y component is non-zero only when the user pushes the peg on the left side of the box or explores the right extremity of the hole (steps (c) and (d)), and remains equal to zero
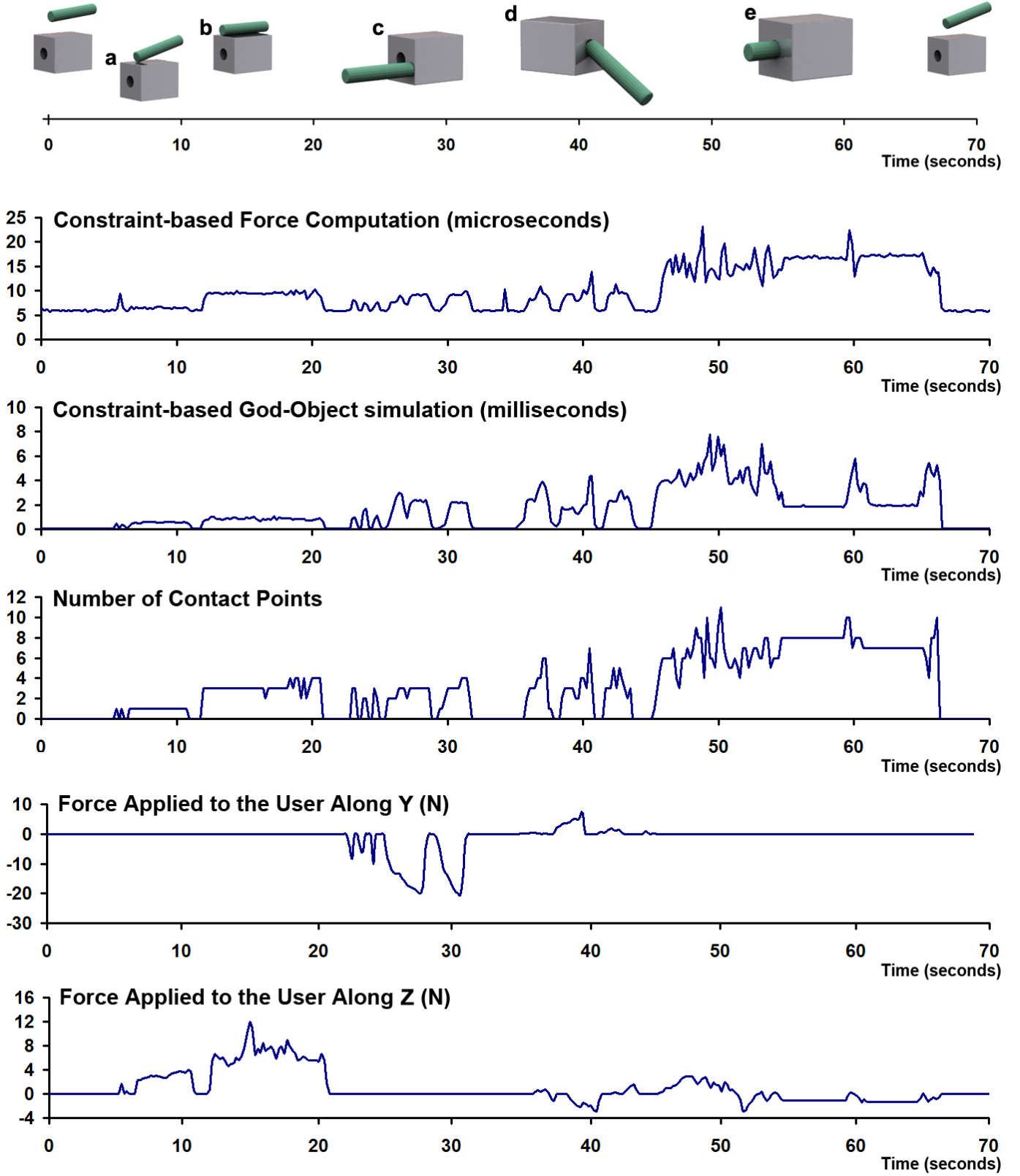
Peg-in-a-hole Benchmark

Fig. 11. **Performance of our approach in the peg-in-a-hole benchmark.** Our method computes a constraint-based force within a few *microseconds*, while a peg configuration update requires only a few milliseconds, which is sufficient to prevent visual lag in the simulation. As expected, the user feels a force in the Y direction only when he pushes the peg on the left side of the box (step (c)) or explores the right extremity of the hole (step (d)). Note how the combination of the collision detection and constraint-based quasi-statics algorithms tend to limit the number of simultaneous contact points during the interaction.

whenever the peg is sliding on the top side of the box or inside the hole. The Z component has high values when the user pushes the peg on the top side of the box, and has little variations when the peg is inseide the hole, due to user movement precision. In other words, the user does not feel any artificial friction force or any artificial sticking during the manipulation (*e.g.* the Y component of the force is never positive during step (c)).

Overall, the combination of continuous collision detection, constraint-based quasi-statics, and constraint-based force computation makes it very easy for the user to accomplish the task, by allowing the peg to slide on the surface of the box and the hole, while providing the user with a high-quality haptic display.

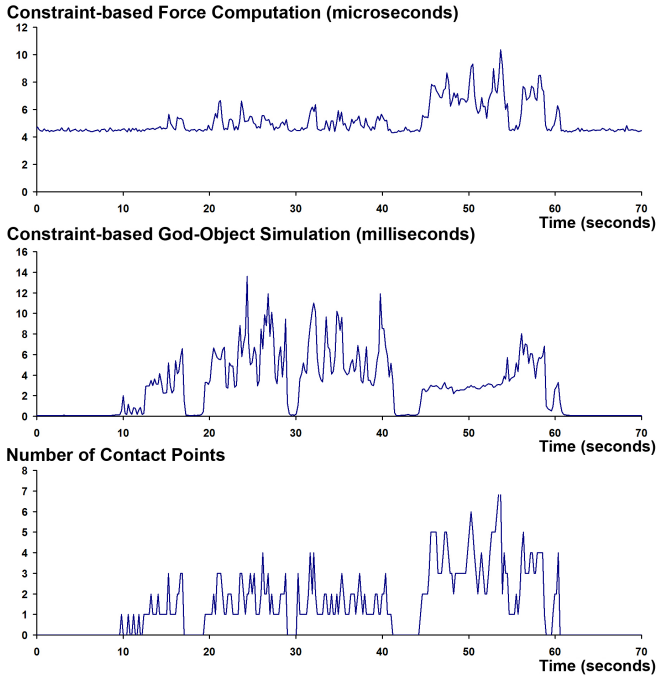## B. Stanford Bunnies Benchmark

Fig. 12. **Performance of our approach in the Stanford Bunny benchmark.** Even in this complex benchmark (27,000 triangles per bunny), our method is able to compute a constraint-based force within a few microseconds. The simulation of the god-object, which includes collision detection and constraint-based quasi-statics computations, is performed in less than 15 milliseconds, which is sufficient to prevent visual lag during the interaction.

The second benchmark involves two Stanford bunnies (27,000 triangles per bunny, see Figure I). One bunny is static, and the second bunny is manipulated by the user. Figure 5 shows several key steps of the interaction: Fig. 5.a shows the ear of the mobile bunny sliding in a ridge of the static bunny; Fig. 5.b demonstrates how the constraint-based god-object simulation provides realistic contacting configurations during the interaction; similarly, Fig. 5.c-d show how our approach is able to provide the user with high-quality haptic display of contacting rigid bodies, where the details of the geometry can be felt by the user.

Figure 12 reports on the performance of our approach during a typical interaction session with the bunnies, which includes

the configurations represented in Figure 5. Again, the force applied to the user is computed within a few microseconds, while an update of the configuration of the mobile bunny, which includes continuous collision detection and constraint-based quasi-statics, is performed within a few milliseconds, resulting in the absence of any visual lag during the interaction.

C. Discussion

1) Benefits: The main benefits of our approach stem from the combination of three key elements:

- **Continuous collision detection** allows the user to feel the details of the geometry of the rigid bodies, and potentially feel the contact between vertices, edges and faces of the contacting objects. Furthermore, the ability to produce visually convincing non-penetrating but tangent contacting configurations (*e.g.* Fig. 5.b) helps us improve the perceived stiffness of the objects [3].
- **Asynchronous updates** of the configuration of the god-object and the force applied to the user help us satisfy the different update rates required by the haptic and the visual displays.
- **Constraint-based quasi-statics** allows the user to slide on the environment obstacles, and haptically feel the reduced motion sub-space resulting from the simultaneous non-penetration constraints, thus providing the user with a realistic haptic display of surfaces, corners, ridges, and object/object contact in general.

Especially, the physically-based computation of the force applied to the user guarantees that no *artificial friction or sticking is felt*, and that no force is applied when the god-object is in free space. This is to be contrasted to what would occur if some kind of virtual coupling was involved in the computation of the force applied to the user. Figure 13 shows such a comparison, in which the god-object (in blue) is constrained to remain above the surface of the obstacle. In the case depicted in Fig. 13.a, where the haptic device (in green) has penetrated the environment, a virtual coupling would attempt to bring the haptic device back to the configuration of the god-object, which would result in an artificial tangential friction applied to the user. As mentioned before, this would degrade the perceived orientation of the surface of the obstacle [4]. In contrast, the constraint-based approach guarantees that the perceived orientation is correct, since the contact forces are always orthogonal to the constraints[3]. Furthermore, in the case depicted in Fig. 13.b, where the user moves away from the obstacle, a virtual coupling would attempt to bring the god-object back to the surface of the obstacle, which would result in a sticky feeling. In this case, however, the constraint-based approach yields the correct force ($\mathbf{F}^c = \mathbf{0}$), since moving away from the obstacle surface satisfies the non-penetration constraint (hence, $\mathbf{a}^c = \mathbf{a}^u$).

Finally, although a complete stability analysis is outside the scope of this paper and left as future work, we believe that

---

[3]Since the constrained acceleration of the god-object $\mathbf{a}^c$ minimizes the kinetic distance $||\mathbf{a}^c - \mathbf{a}^u||_{\mathbf{M}}$ to the unconstrained acceleration $\mathbf{a}^u$ among the possible accelerations, it is such that $(\mathbf{a}^c - \mathbf{a}^u)^T \mathbf{M} \mathbf{a}^c = 0$, which implies that $(\mathbf{F}^c)^T \mathbf{a}^c = 0$.
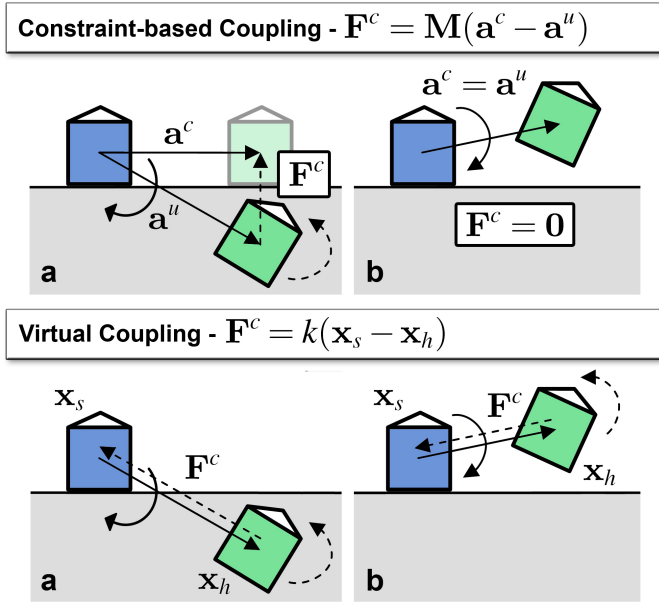
Fig. 13. **Benefits of the constraint-based approach**. The constraint-based approach introduced in this paper allows us to remove force artifacts typically found in previous methods (see Section VII).

the asynchronous constraint-based approach helps us improve the stability of the interaction. Indeed, it can be shown that the simulation of the god-object is purely dissipative, i.e. that the force $\mathbf{F}^u = \mathbf{M}\mathbf{a}^u$ applied to the god-object is such that

$$(\mathbf{F}^u)^T \mathbf{a}^c \leqslant (\mathbf{F}^u)^T \mathbf{a}^u.$$

Thus, the non-penetration constraints can only dissipate the energy transmitted to the god-object[4]. Our tests have shown that the user is able to *e.g.* release the handle of the haptic device while the peg is inside the hole (cf Figure 11, step (e)).

2) Limitations: Our approach has two main limitations:

- **Linearized constraints:** in order to efficiently compute the quasi-statics of the god-object and the constraint-based force applied to the user, the non-penetration constraints are linearized. This might reduce the quality of the force applied to the user when a large discrepancy between the configurations of the god-object and the haptic device occurs. It would be interesting to investigate some more sophisticated force computation methods to address this problem, involving for example an implicit formulation of the non-penetration constraints.

- **Potentially low update rate of the set of constraints:** we do not guarantee that our approach is able to update the set of non-penetration constraints at 1000 Hz. This might lead to missing some high-frequency details when the user slides rapidly on the surface of the environment obstacles.

The potentially low update rate of the set of constraints is the main reason for the separation of the god-object simulation

and the constraint-based force computation into asynchronous processes, in our approach and several previous ones (*e.g.* [16], [25]). Because the complexity of any collision detection method which reports all the contacting features is output-dependent, however, it seems arguable that, whichever collision detection method is used, it will always be possible to find a scenario such that the time required to determine all the contact points will take more than one millisecond. We have thus preferred to rely on a god-object simulation method which offers precise interaction with rigid bodies and, especially, precisely contacting configurations. Although this might limit the rate at which the set of non-penetration constraints is updated (sometimes as low as 70 Hz in the Stanford bunnies benchmark, and about 300 Hz on average), this approach allows us to compute a constraint-based force consistent with the current set of simultaneous constraints at extremely high rates (always higher than 80,000 Hz in the Stanford bunnies benchmark). Furthermore, it should be emphasized that the constraint-based computations performed in the constraint-based coupling loop *implicitly include some collision detection*. Returning to the example depicted in Figure 4, it can be seen that if, between two updates of the set of non-penetration constraints, the haptic device switches from a state where all currently known non-penetrating constraints are satisfied (in which case $\mathbf{F}^c = \mathbf{0}$) to one where at least one of the currently known non-penetrating constraint is not satisfied (in which case $\mathbf{F}^c \neq \mathbf{0}$), the user will feel this collision. In summary, collision detection is implicitly performed in the constraint-based coupling loop, for the current set of non-penetration constraints, at extremely high rates.

VIII. CONCLUSION AND FUTURE WORK

This paper has introduced a new method for six degree-of-freedom haptic display of rigid bodies, which generalizes the classical three degree-of-freedom god-object method introduced by Zilles and Salisbury [2]. As in their initial approach, the god-object is able to contact and slide on the environment obstacles without penetrating them, and the forces applied to the user are orthogonal to the non-penetration constraints. Our approach has been successfully tested on the classically difficult peg-in-a-hole benchmark and on some more complex models - two Stanford bunnies with 27,000 triangles each. We have shown that our method is able to provide a high-quality haptic display of contacting rigid bodies in both cases. We also demonstrate that the approach is compatible with the generation of textures and force shading. In this case our constraint-based approach ensures that no force artifacts are felt by the user.

There are several directions for future work. Besides addressing the limitations described above, we would like to extend our approach to multiple dynamic objects (although it can be argued that quasi-static interaction is preferable for the simulation of many tasks, as few manipulation tasks seem to require using the inertia of the manipulated object to accomplish the task). One possible direction to do this could be to generalize the approach suggested by Niemayer and Mitra [7] to six degree-of-freedom haptic interaction. Finally, we

---

[4]The proof is straightforward. Indeed, $(\mathbf{F}^u)^T(\mathbf{a}^c - \mathbf{a}^u) = (\mathbf{a}^u)^T \mathbf{M}(\mathbf{a}^c - \mathbf{a}^u) = -||\mathbf{a}^c - \mathbf{a}^u||_{\mathbf{M}}^2 + (\mathbf{a}^c)^T \mathbf{M}(\mathbf{a}^c - \mathbf{a}^u)$. Since $(\mathbf{a}^c - \mathbf{a}^u)^T \mathbf{M}\mathbf{a}^c = 0$ (see footnote 3), $(\mathbf{F}^u)^T \mathbf{a}^c \leqslant (\mathbf{F}^u)^T \mathbf{a}^u$. Note that we use the product of the force and the acceleration because we deal with the quasi-static case. This is the equivalent of the product of the force and the velocity used in typical analyses.

plan to investigate actual industrial scenarios such as virtual prototyping and assembly tasks.

REFERENCES

[1] M. Ortega, S. Redon, and S. Coquillart, "A six degree-of-freedom god-object method for haptic rendering of rigid bodies," in *Proc. of IEEE International Virtual Reality Conference*, march 2006.

[2] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 1995.

[3] M. A. Srinivasan, G. L. Beauregard, and D. L. Brock, "The impact of visual information on the haptic perception of stiffness in virtual environments," in *Proc. of ASME Winter Annual Meeting*, November 1996.

[4] W. L. Sachtler, M. R. Pendexter, J. Biggs, and M. A. Srinivasan, "Haptically perceived orientation of a planar surface is altered by tangential forces," in *Proc. of Fifth Phantom User's Group Workshop*, 2000.

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

[6] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Computer Graphics (ACM SIG-GRAPH'97 Proceedings)*, August 1997.

[7] G. Niemeyer and P. Mitra, "Dynmic proxies and haptic constraints," in *Workshop on Multi-point Interaction in Robotics and Virtual Reality*, 2004.

[8] K. Salisbury and C. Tarr, "Haptic rendering of surfaces defined by implicit functions," in *Proc. of ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1997.

[9] L. Kim, A. Kyrikou, G. S. Sukhatme, and M. Desbrun, "An implicit-based haptic rendering technique," in *Proc. of IEEE Intelligent Robots and System*, 2002.

[10] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Computer Graphics (ACM SIGGRAPH Proceedings)*, 1999.

[11] D. E. Johnson, P. Willemsen, and E. Cohen, "Six degree-of-freedom haptic rendering using spatialized normal cone search," in *IEEE Transactions on Visualization and Computer Graphics, Volume 11, Issue 6, Nov.-Dec. 2005 Page(s):661 - 670*, 2003.

[12] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha, "Six degree-of-freedom haptic display of polygonal models," in *Proc. of IEEE Visualisation*, 2000.

[13] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, "Six-degree-of-freedom haptic display using localized contact computations," in *Proc. of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002.

[14] M. A. Otaduy and M. C. Lin, "Sensation preserving simplication for haptic rendering," in *ACM Transactions on Graphics*, 2003.

[15] S. Hasegawa and M. Sato, "Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects," in *Proc. of IEEE EUROGRAPHICS'04*, 2004.

[16] D. Constantinescu, S. E. Saludean, and E. A. Croft, "Haptic rendering of rigid body collisions," in *Proc. of 12th IEEE International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004.

[17] M. Wan and W. A. McNeely, "Quasi-static approximation for 6 degrees-of-freedom haptic rendering," in *Proc. of IEEE Visualisation, pp. 257-262*, 2003.

[18] D. Nelson, D. Johnson, and E. Cohen, "Haptic rendering of surface-to-surface sculpted model interaction," in *Proc. of 8th ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1999.

[19] P. Berkelman, R. Hollis, and D. Baraff, "Interaction with a realtime dynamic environment simulation using a magnetic levitation haptic interface device," in *Proc. of IEEE International Conference on Robotics and Automation*, 1999.

[20] M. A. Otaduy and M. C. Lin, "Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration," in *Proc. of IEEE World Haptics Conference*, 2005.

[21] S. Redon, A. Kheddar, and S. Coquillart, "Fast continuous collision detection between rigid bodies," in *Computer Graphics Forum 21 (3) (IEEE Eurographics Proceedings)*, 2002.

[22] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," in *In Computer Graphics (Proc. ACM SIG-GRAPH), volume 23, pages 223-232*, 1989.

[23] K. F. Gauss, "Uber ein neues allgemeines grundgesatz der mechanik," in *Journal f'ur die Reine und Angewandte Mathematik, 4, pp 232-235*, 1829.

[24] D. R. Wilhelmsen, "A nearest point algorithm for convex polyhedral cones and applications to positive linear approximations," in *Mathematics of computation, 30, pp 48-57*, 1976.

[25] W. R. Mark, S. C. Randolph, M. Finch, J. M. V. Verth, and R. M. Taylor II, "Adding force feedback to graphics systems: Issues and solutions," in *Proc. of ACM SIGGRAPH'96*, August 1996.

[26] H. B. Morgenbesser and M. A. Srinivasan, "Force shading for shape perception in haptic virtual environments," in *Touch Lab report 4, MIT*, December 1996.

[27] B. Phong, "Illumination for computer generated pictures," in *Communications of the ACM, 18(6), pp 311-317*, June 1975.

[28] A. S. M. A. Otaduy, N. Jain and M. C. Lin, "Haptic display of interaction between textured models," in *Proc. of IEEE Visualization Conference, pp. 297-304. Austin, Tx.*, 2004.

[29] M. A. Otaduy and M. C. Lin, "A perceptually-inspired force model for haptic texture rendering," in *Proc. of the ACM Symposium on Applied Perception in Graphics and Visualization, pp. 123-126. Los Angeles, Ca.*, 2004.

[30] J. Siira and D. Pai, "Haptic textures - a stochastic approach," in *Proc. of IEEE International Conference on Robotics and Automation*, 1996.

[31] C. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient point-based rendering techniques for haptic display of virtual objects," in *Presence 8(5)*, 1999, pp. 477–491.

[32] D. K. Pai, K. V. D. Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau, "Scanning physical interaction behavior of 3d objects," in *Proc. of ACM SIGGRAPH*, 2001.

[33] M. D. R. Minsky, "Computational haptics: The sandpaper system for synthesizing texture for a force-feedback display," in *PhD Thesis, MIT*, June 1995.

[34] J. F. Blinn, "Simulation of wrinkled surfaces," in *Computer Graphics, Vol. 12 (3), pp. 286-292 ACM SIGGRAPH*, August 1978.

[35] S. Kim, S. Hasegawa, Y. Koike, and M. Sato, "Tension based 7-dof force-feedback device: Spidar g," in *Proc. of IEEE Virtual Reality Conference*, March 2002.

[36] N. Tarrin, S. Coquillart, S. Hasegawa, L. Bouguila, and M. Sato, "The stringed haptic workbench : a new haptic workbench solution," in *Proc. of IEEE EUROGRAPHICS*, September 2003.