A 6-DoF Zero-Order Dynamic Deformable Tool for Haptic Interactions of Deformable and Dynamic Objects

Haiyang Ding^{1(✉)} and Shoichi Hasegawa^{2,3,4}

¹ Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology, Tokyo, Japan
haiyang.d.aa@m.titech.ac.jp

² Precision and Intelligence Laboratory, Tokyo Institute of Technology, Tokyo, Japan
hase@pi.titech.ac.jp

³ Independent Administrative Corporation, Tokyo, Japan

⁴ Japan Science and Technology Agency (JST), Kawaguchi, Japan

Abstract. Continuous collision detection is required for haptic interactions with thin and fast-moving objects. However, previous studies failed to eliminate the force artifacts caused by the tool's inertia. In this paper, we propose a multi-sphere proxy method for a 6-DoF deformable virtual tool with continuous collision detection. We use Zero-order Dynamics to avoid force artifacts caused by the tool's inertia. In addition, we eliminate the “tunneling” problem introduced by the use of Zero-order Dynamics. As a result, we support fast motions of both the tool and the virtual object with sphere-mesh-level contacts and real-time simulation. Stability is guaranteed via a position-based dynamics simulator and an optional multi-rate architecture.

Keywords: Haptic rendering · Continuous collision detection · Deformable objects

1 Introduction

Haptic interaction requires accurate collision detection in order to render the contact force between the virtual tool and the virtual environment. The “tunneling” problem, also called the pop-through effect, caused by using discrete collision detection can be avoided by using continuous collision detection (CCD).

Garre et al. [3] presented a deformable 6-DoF tool which was able to interact with deformable objects using CCD. However, they used bidirectional viscoelastic coupling between the device and the tool, which introduced force artifacts

---

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58147-3_27) contains supplementary material, which is available to authorized users.

caused by the force of the tool’s inertia. These artificial forces reduce rendering transparency, and fast motion input from the device may trigger stability problems when interacting with heavy objects.

To the contrary, the traditional proxy method [11] can directly manipulate the proxy position without force artifacts. This simulation method is called Zero-order Dynamics (ZoD) [6].

In this paper, we propose a multi-sphere proxy (MSP) model to achieve 6-DoF haptic interactions with ZoD. In addition, this method is also able to perform CCD by using the method from [2] to handle collisions between the proxy sphere and triangle meshes.

However, ZoD may induce the “tunneling” problem if the collisions are not identified during the constraint computation. In this paper, we perform an additional CCD to solve this problem. Therefore, constraints, such as deformation, can be performed correctly without causing the “tunneling” problem. We list our contributions below:

- An MSP model which enables 6-DoF haptic interactions with dynamic and deformable objects through sphere-mesh-level CCD contacts;
- A ZoD simulation of a deformable 6-DoF haptic tool which eliminates the force artifacts caused by the tool’s inertia as well as the “tunneling” problem.

2 Background and Related Works

2.1 Haptic Rendering with Zero-Order Dynamics

ZoD [6] was proposed to describe the simulation of the traditional virtual proxy method [11]. This method uses a virtual finite-radius sphere, i.e., the proxy, to represent a 3-DoF virtual tool. When the device moves, the proxy tries to follow its path via iterative collision detections. If contact occurs, the proxy finds the closest position to the device constrained by contacts. If there is no contact, the proxy tracks the device perfectly. The time integration has no mass or velocity involved.

When considering the dynamics of the 6-DoF haptic tool, many studies use second-order dynamics [3, 4] with virtual coupling [10], thus producing force artifacts. Meanwhile, the constraint-based [9] and configuration-based optimization [12] methods can be used to remove force artifacts. However, these methods are limited to rigid tools. Mitra et al. [6, 7] used first-order dynamics but position constraints must be transferred into velocity constraints making contact forces difficult to compute.

2.2 Haptic Rendering with Continuous Collision Detection

The traditional virtual proxy method [11] checks just the collisions between the moving proxy and static virtual objects. Therefore, if a dynamic thin virtual object is moving quickly or deforming greatly during one time step and pass through the proxy position, the collision will not be detected.

Ding et al. [2] proposed a method utilizing triangle-proxy CCD and Proxy Pop-out processes to solve this problem. Triangle-proxy CCD computes the CCD between the triangle and the proxy by solving the coplanar condition between the moving triangle and the proxy center. After that, the Proxy Pop-out carries our iterative discrete collision detections between multiple triangular meshes and the proxy sphere. These processes are used to secure the start of the proxy to avoid the “tunneling” problem.

For the CCD used in 6-DoF haptic rendering, Ortega et al. [9] used a constraint-based god-object method for rigid-rigid haptic interactions. Garre et al. [3, 4] can simulate deformation on both the tool and the object. Other haptic rendering approaches can be found in the survey [10].

3 Proposal

3.1 Multi-sphere Proxy Model for a Virtual Tool

As described in Sect. 2.2, CCD between the proxy and dynamic triangular models can be computed using a sphere proxy[2]. Here, we extend the method used in [2] via the MSP model.

The Oriented Particles (OP) [8] method uses ellipsoid particles to simulate solid deformations. We directly use the particles of the OP model as our MSP model. The particle of the virtual tool is called the tool particle, and the particle representing the device is called the device particle. Since we apply CCD and proxy method described in [2] for each tool particle, we use spherical particles instead of ellipsoids for better performance. However, since ellipsoids can be used to represent a more precise contact model compared to spheres, it will be interesting to conduct further research using ellipsoids.

Since the number of tool particles is related directly to both collision accuracy and simulation speed, collision accuracy and system efficiency should be balanced. In this paper, we use the same method as [8] to create the OP model. An example is shown in Fig. 3.

This method is similar to that employed in [1] and [12], which also utilize spheres. However, their methods cannot handle CCD contact between the sphere and the triangular mesh, and method in [1] cannot avoid force artifacts caused by the tool’s inertia.

3.2 Six Degree-of-Freedom Haptic Rendering

Force and Torque Feedback. We calculate the force and torque feedback from the discrepancy between the virtual tool and the device. As shown in Fig. 1, we first calculate force and torque of each pair of tool and device particles. Only the particles that have collided are considered in the calculation. Next, we sum all the forces and all the torques, respectively. Finally, we divide each of these sums by the total number of tool particles. We use translational and rotational springs to adjust the feedback magnitude.

Fig. 1. Force and torque calculation of the multi-sphere Proxy model. (a) The start. (b) The deformable tool encounters a contact. (c) The force $\mathbf{f}_i$ and $\mathbf{f}_j$ and, (d) the torque $\mathbf{r}_i \times \mathbf{f}_{i,t}$ and $\mathbf{r}_j \times \mathbf{f}_{j,t}$ of tool particle i and j , respectively.

Haptic Contact Force Computation. The contact force applied to the virtual object is computed using the method in [2], which uses a distributed point-like contact force applied as an external force of PBD to the corresponding OP particle of the virtual object.

Multi-rate Architecture. To improve interaction stability, we use a multi-rate architecture. First, we simply synchronize all the tool particle positions from the physics thread to the haptic thread. The feedback force and torque are calculated in the haptic thread. However, the delay caused by synchronization may induce “dragging” forces. Therefore, we also synchronize the collision information (collision occurs or not) to eliminate the force artifacts when no contact occurs.

3.3 Zero-Order Dynamic Deformable Haptic Tool

As introduced in Sect. 2.2, the traditional proxy method [11] uses ZoD to avoid force artifacts. We achieve ZoD by manipulating the position of the tool particle directly with a position-based haptic constraint.

Haptic Constraint. The haptic constraint helps us to apply the haptic device input to the virtual tool by computing current position of the device particle as the goal position. The computation is as follows:

$$\mathbf{g}_i = Q_{dev} \mathbf{o}_i + \mathbf{d}_{dev}. \quad (1)$$

Here, $\mathbf{o}_i$ represents the tool particle’s original barycentric coordinates before deformation, while Q_{dev} and $\mathbf{d}_{dev}$ represent for the device’s orientation matrix (3×3) and position, respectively. The variable with a bold font refers to a 3×1 vector.

After the goal position is computed, we perform the traditional proxy method [11] to update tool particles to the goal positions, i.e., the positions of device particles (shown in Fig. 2(b)).

The “Tunneling” Problem Caused by Constraints. Since we use a stiff haptic constraint in PBD, it cancels the effect of the other constraints. To solve this problem, we apply the haptic constraint only once before the other constraints. Therefore, the other constraints, e.g., the deformation constraint, can be performed correctly.

However, if we perform the constraint calculation, the result may violate the collision constraints and cause the “tunneling” problem (shown in Fig. 2(c)). Unfortunately, it is impossible to identify all of the collision constraints ahead in the proxy method collision detection we executed when applying the haptic constraint since they have different starts and goals.

Solving the “Tunneling” Problem. To solve the “tunneling” problem caused by the constraint calculation, our proposal is to run an additional CCD, which is the same as the proxy method, for the constraint calculation (shown in Fig. 2(d)) to revise the particle positions. In order to do so, we record the tool particle positions before and after the PBD constraint calculation. After that, we perform the proxy method between the recorded start and end. The whole progress is shown in Fig. 2.

Fig. 2. Simulation of a zero-order dynamic deformable haptic tool

3.4 Simulation of a Zero-Order Dynamic Deformable Haptic Tool

Our simulation loop of the physics thread is executed as follows:

1. Physics thread start:
2. Simulate the virtual tool:
  - (a) Perform triangle-CCD and Proxy Pop-out for tool particles; (Sect. 2.2)
  - (b) Calculate and apply haptic constraint; (Sect. 3.3)
  - (c) Perform the traditional proxy method for each tool particle;
  - (d) Record current tool particle positions;
  - (e) Perform and record the result of other PBD constraint iterations;
  - (f) Perform proxy CCD with the recorded start and end; (Sect. 3.3)

- (g) Update the tool particle position;
  - (h) Calculate haptic contact forces; (Sect. 3.2)
3. Apply haptic contact forces as external forces to the contact object;
  4. Simulate the other virtual objects.

Fig. 3. Evaluation of interactions. Left: the recorded tool position, feedback force, and torque are presented with screenshots of a deformable hand interacting with a yellow curtain. Right: the hand model and its tool particles with the proxy radius. (Color figure online)

4 Evaluations

In this section, we will introduce two evaluations of our method. The evaluations were performed on an Intel i5-7300 4-core CPU PC without GPU. A Spidar-G6 was used as the haptic device. We also provide a supplementary video.

4.1 The Evaluation of Interactions and Efficiency

In this evaluation, we used a deformable 1 kg hand model [5] as the tool to interact with a double-faced, 8-m, 200 g curtain model. The curtain model contained 1089 vertices and 4096 triangles, while each vertex was modeled with one OP particle and connected with another eight nearby particles. The hand model had 3919 vertices and 3906 triangles with 37 OP particles (the placement of the tool particles can be further optimized). Only shape-matching and distance constraints were simulated for the two models. The PBD iteration counts were two and four for the curtain and the hand models, respectively. The feedback springs were both set to 20 N/m. No damping of virtual coupling was used.

The result is displayed in Fig. 3. A series of interactions were performed, including tapping and pressing, on the curtain model. From Fig. 3, we can see

that the tool moved freely in the air without the force caused by the inertia. The system was stable even fast motions were inputted with strong impacts while the tool and the device deformation were performed correctly. The whole simulation took 2.5 s. The data was collected from the haptic thread which was running at 1 kHz, while one time-stamp indicated one loop of the calculation.

The simulation ran at an average rate from 45 to 59 FPS, depended on the collision number. The calculation of the PBD constraints took 4 ms. The CCD, including triangle-proxy CCD and Proxy Pop-out, required 3 ms, while the two runs of the proxy method required 8 to 12 ms depending on the collision count. We used a basic pruning method which ignored far-off meshes (at least 1 m from every tool particle). The computational efficiency was highly related to the number of tool particles; therefore, we also ran a test of this scene using tool particles numbering from 1 to 50 (illustrated in Fig. 4(a)).

4.2 The Evaluation of Multi-rate Haptic Rendering

We conduct a comparison between multi-rate and single-rate haptic rendering to evaluate the force artifacts caused by multi-rate simulation. In this evaluation, the device slid along a virtual horizontal plane using the same hand model of the first evaluation. The results in Fig. 4(b) and (c) show that the horizontal “dragging” forces (forces on X and Z axis) arise only in the multi-rate case as the device moves. During our tests, the magnitude of the force artifacts was proportional to the computation time of the physics simulation.

Fig. 4. Evaluation of efficiency (a) and Comparison Between Multi-Rate (b) and Single-Rate (c) Haptic Rendering. The efficiency evaluation utilized the average computation time for the simulation and proxy method using 1 to 50 tool particles. The label with “*” indicates collisions are occurred in the simulation while others are not.

5 Conclusion, Limitations, and Future Works

We propose a 6-DoF deformable tool that can interact with another deformable virtual object. The contact is handled between spherical proxies and triangular

meshes. The force artifacts caused by the tool's inertia were eliminated, and tool deformation was performed without the "tunneling" problem. Our simulation is stable even with fast motion input using a large feedback spring. We also provide a multi-rate haptic rendering option which improves the stability but induces other force artifacts.

Compared to other studies, our system uses a less accurate contact tool model compared to the one in [3], in which the tool has 1441 triangular meshes for collision detection (not sure about the deformable object). The hand model we used in the evaluations does not have a rigid-body or joint constraint; therefore, the physics behavior of the hand model is less accurate than that used in [3]. Also, as we represent the tool with spherical particles, objects that are thinner than the gap between particles may result in penetrations. In the future, we plan to find solutions to cover this gap and use ellipsoids to handle accurate contacts.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number 17H01774. Here, we are also grateful to the support of Springhead physical engine and the developers.

References

1. Cirio, G., Marchal, M., Otaduy, M.A., Lécuyer, A.: Six-of haptic interaction with fluids, solids, and their transitions. In: 2013 World Haptics Conference (WHC), pp. 157–162. IEEE (2013)
2. Ding, H., Mitake, H., Hasegawa, S.: Continuous collision detection for virtual proxy haptic rendering of deformable triangular mesh models. *IEEE Trans. Haptics* **12**(4), 624–634 (2019)
3. Garre, C., Hernández, F., Gracia, A., Otaduy, M.A.: Interactive simulation of a deformable hand for haptic rendering. In: 2011 IEEE World Haptics Conference, pp. 239–244. IEEE (2011)
4. Garre, C., Otaduy, M.A.: Haptic rendering of objects with rigid and deformable parts. *Comput. Graph.* **34**(6), 689–697 (2010)
5. gotferdom: Free hand 3D model (2018). <https://www.turbosquid.com/3d-models/hand-hdri-shader-3d-model-1311775>. Accessed 25 Jan 2020
6. Mitra, P., Niemeyer, G.: Dynamic proxy objects in haptic simulations. In: 2004 IEEE Conference on Robotics, Automation and Mechatronics, vol. 2, pp. 1054–1059. IEEE (2004)
7. Mitra, P., Niemeyer, G.: Haptic simulation of manipulator collisions using dynamic proxies. *Presence Teleoperators Virtual Environ.* **16**(4), 367–384 (2007)
8. Müller, M., Chentanez, N.: Solid simulation with oriented particles. *ACM Trans. Graph.* **30**, 92 (2011)
9. Ortega, M., Redon, S., Coquillart, S.: A six degree-of-freedom god-object method for haptic display of rigid bodies. In: IEEE Virtual Reality Conference (VR 2006), pp. 191–198. IEEE (2006)
10. Otaduy, M.A., Garre, C., Lin, M.C.: Representations and algorithms for force-feedback display. *Proc. IEEE* **101**(9), 2068–2080 (2013)
11. Ruspini, D.C., Kolarov, K., Khatib, O.: The haptic display of complex graphical environments. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 345–352 (1997)

12. Wang, D., Tong, H., Shi, Y., Zhang, Y.: Interactive haptic simulation of tooth extraction by a constraint-based haptic rendering approach. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 278–284. IEEE (2015)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.