Haptic Implementation on deformation models based on Oriented Particles

Haiyang Ding[1], Hironori Mitake[2] and Shoichi Hasegawa[3]

***Abstract*— We propose a haptic rendering method for deformable models based on oriented particles. Oriented Particles(OP) is a fast and robust deformation method which is better fitted for multi object and robust deformation simulation systems. Our research focus on implementing haptic interaction for OP. Since the configuration of particles can be treated as the bounding volume of their belonging vertices, we propose the proxy collision detection by a two-layer collision procedure. Besides, haptic input is introduced into the simulation system as an external force and projected to the particles by the proportion of the distance. Our contribution is a basic haptic interaction implementation of OP. Thanks to the fast simulation of OP, our system is able to handle multiple objects with fast and robust haptic interaction.**

I. INTRODUCTION

The dynamic simulation of haptic interaction with deformable objects has been investigated for more than two decades. Most researchers focus on high fidelity representation in a small scale simulation. They choose Finite Element Model(FEM) as the prior method for deformation. Since FEM is a discretization of continuous mechanics, material parameters can be used to presented realistic deformation. However, the computation time of general FEM is still a big issue for large scene or multi-objects simulation systems with collisions and haptics. Mostly because the calculation of updating the stiffness matrix is too expensive. Besides, using invariant stiffness matrix without updating causes unignorable errors while performing large deformation. Therefore, simplification or GPU computation is necessary for those systems.

On the contrary, Oriented Particles(OP) is based on a geometry matching method, which is called "shape-matching". For this reason, the stiffness parameters can not be derived from material parameters, which means creating a real-object deformation simulation becomes difficult. Due that OP uses an iterative solver for multiple stiffness constraints, OP does not need to calculate the stiffness matrix, so it is faster. In the meantime, OP holds good properties such as robust deformation, rough but fast collision, robust skinning and also it is easy to implement. Although no theoretically comparison on speed between FEM and OP is proposed, in practice, OP is able to simulate large scene of multi-deformable-objects together with collision detection in real-time. Thus, we are considering a haptic implementation based on OP will be useful to create multi-object or large scene simulation systems, even user interactions with large robust deformation can be handled correct and fast without GPU. Next, we will describe our work in the following section.

[1]Haiyang Ding, Tokyo Institute of Technology, Department of Computational Intelligence and Systems Science teikaiyo@haselab.net

[2]Hironori Mitake, Tokyo Institute of Technology, Precision and Intelligence Laboratory. mitake@pi.titech.ac.jp

[3]Shoichi Hasegawa, Tokyo Institute of Technology, Precision and Intelligence Laboratory; Independent Administrative Corporation; Japan Science and Technology Agency(JST); hase@pi.titech.ac.jp

II. RELATED WORK

Over the past decades, much attention has been devoted to research on haptic simulation with deformable objects. Especially for complex model and multiple objects, Barbic[1] succeeded in presenting a two-objects real-time haptic interaction. He achieved it by a reduced FEM method, named St.Venant-Kirchhoff Material. Galoppo[2] introduced a haptic simulation environment that can handle multiple objects. These objects are simulated by the rotationally invariant FEM method, but it is still inevitable to use the GPU to achieve real-time simulation.

About OP, Muller[3] introduced OP as a deformable model of sparse particle structure arranged on vertices based on shape-matching. The consequence of introducing orientation is that less particles are necessary to present same behavior compare to normal particles.

III. METHOD

We use the proxy method[4] for haptic rendering. As OP calculate collision detection on particles, we first tried a simply implementation of haptic rendering by considering the proxy as another particle. Since particles are not able to present model surface precisely, this implementation gives a rough feedback or uneven feeling compare with the visual mesh. Then, if we implement surface mesh contact with proxy directly, the problem will be solved. However, in this case, additional collision detection for proxy with global meshes is needed. After those trials, we come to the two-level collision detection algorithm.

A. Two-level Collision Detection

In OP, one particle is arranged to a certain area grouping vertices to control them like a skeleton, and also being used for collision detection. Thus, the particle radius normally covers most of the vertices in the surface mesh. This configuration is similar to the fast collision detection method named Bounding Volume Hierarchy(BVH), so we come up to the idea to create a similar structure. We bind the meshes to the particle which hold the mesh vertex. Then the collision detection is appended into two steps:first is the normal process of OP collision detection, in which we

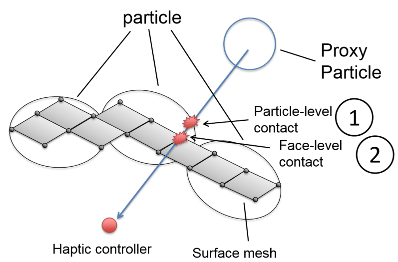

Fig. 1. The collision detection for the first layer is between proxy particle to model particles. The second layer is between proxy to local surface meshes

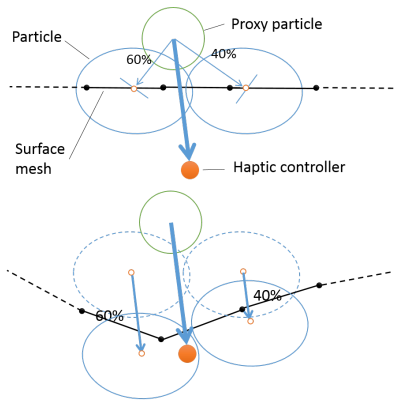

Fig. 2. Haptic input force is projected to particles by the proportion of distance from proxy particle to nearby particles.

consider proxy as a special particle taking part in the particle detection process. Then, as we found which particle collides with proxy particle, additional detection for proxy with the meshes binding in the collied particle is played locally same as the proxy method. Here the particle is treated as bounding volume of certain vertices. As long as any vertex escapes from particle radius, our collision detection will not miss. This procedure is present in fig. 1.

B. Haptic Implementation

Our research starts from three-degree-of-freedom (3-DOF) haptic interaction.

For OP is simulated based on Position Based Dynamics(PBD)[5], we calculate haptic force input as a PBD external force by the penalty method[6]. As in OP, vertices are controlled by particles, so haptic force should be projected to particles. We propose a distance proportion force projection. The particle which is closer to the proxy particle is projected more force than others. Our example is shown in fig.2.

IV. Result

We create a scene contains four reduced stanford bunnys. Each model has 458 vertices arranged with 80 particles. Then we are able to interact with them by the haptic device Spidar-G. This demonstration is shown in fig.3.

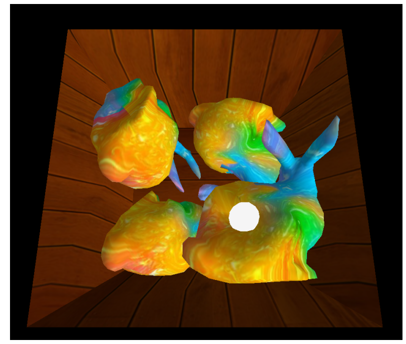

Fig. 3. Bunnies are dropped in a basket. The white ball is the proxy controlled by user.

V. Discussion

Our proposal achieves a basic implementation of haptic on OP simulation system. By taking advantage of OP particle configuration, we create a two-level structure for haptic collision detection. The idea comes from the BVH method. Besides, a projection from haptic input to force applied on particle is presented, which gives a way to transfer user input to particles and to virtual model.

However this implementation is far from completed. First, as we pointed out in the last of collision detection, vertices which been stretched out of particle radius will cause miss in the haptic contact. We are working on solving it by dynamically updating the particle radius, which will keep cover all vertices. Second, the haptic input is introduced to the simulation system as ”external force”. Thus, according to PBD simulation, haptic force is calculated as the explicit integration which may cause unstable interaction with strong input or low stiffness model. A stable way of haptic rendering for PBD is being established.

VI. Future Work

The system is better to show interactions with multiple objects. More completed and advanced haptic rendering implementation such as 6-degree-of-freedom, fiction etc. are being developed. Besides, further research on modeling realistic physical simulation by OP is under exploration.

References

[1] Barbic, J., & James, D. L. (2008). Six-dof haptic rendering of contact between geometrically complex reduced deformable models. Haptics, IEEE Transactions on, 1(1), 39-52.

[2] Galoppo, N., Tekin, S., Otaduy, M. A., Gross, M., & Lin, M. C. (2007). Interactive haptic rendering of high-resolution deformable objects. In Virtual Reality (pp. 215-223). Springer Berlin Heidelberg.

[3] Muller, M., & Chentanez, N. (2011, August). Solid simulation with oriented particles. In ACM transactions on graphics (TOG) (Vol. 30, No. 4, p. 92). ACM.

[4] Ruspini, D. C., Kolarov, K., & Khatib, O. (1997, August). The haptic display of complex graphical environments. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (pp. 345-352). ACM Press/Addison-Wesley Publishing Co..

[5] Muller, M., Heidelberger, B., Hennix, M., & Ratcliff, J. (2007). Position based dynamics. Journal of Visual Communication and Image Representation, 18(2), 109-118.

[6] McNeely, W. A., Puterbaugh, K. D., & Troy, J. J. (2005, July). Six degree-of-freedom haptic rendering using voxel sampling. In ACM SIGGRAPH 2005 Courses (p. 42). ACM.