Haptic Deformation using Graphics Hardware and kd-trees

Beom-Chan Lee* Jong-Phil Kim† Jeung-Chul Park‡ Kwan H. Lee§ Jeha Ryu¶
Dept. Mechatronics, Gwangju Institute of Science and Technology(GIST), Republic of Korea

1 Introduction

In this paper, we describe a haptic deformation algorithm using graphics hardware and spatial partitioning data structure. Since object-based collision detection methods are generally dependent on the geometrical structure of objects, these are not fast enough for deformable objects. Thus Z-buffer depth comparison using graphics hardware is used for detecting collision between a haptic probe and an object. In order to determine what region to deform in response to a user's actions, we adopted a nearest neighbor search algorithm based on a kd-tree structure [Moore. 1991]. At the same time, the reaction force to be presented to the user is calculated in the haptic rendering loop. To compute response force for object's deformation, mass-spring methods are utilized. Where masses are assigned to vertices and a set of spring are allocated to connect vertices.

2 Haptic Deformation Algorithm

The haptic deformation is composed of three procedures: collision detection, deformation, and force computation.

The collision detection process is performed on the graphics hardware by using Z-buffer depth information. Haptic interaction in such a point based mode generally occurs between the haptic probe and a very small portion of the virtual environment. Thus, to perform collision detection, only the local geometry information near the probe is required. In order to generate this information with the graphics hardware, we used virtual cameras which see a portion of virtual object corresponding to the haptic probe [Kim et al. 2004]. With local depth information with respect to haptic interaction point, a collision between the probe's position and a portion of the virtual object can be easily detected through depth comparison.

In order to determine deformable vertices, we adopted a nearest neighbor search algorithm based on a kd-tree structure that is a data structure for storing a finite set of points in a k-dimensional space; essentially a binary tree [Moore. 1991]. Given a collision point and using the nearest neighbor algorithm, it is relatively easy to generate a list of the most proximal vertices. Using this technique a region that will be subject to deformation is calculated at the stage of force computation. The region is simply defined as all vertices within a given distance from the collision point between the probe position and the virtual object. Essentially, the nearest neighbor algorithm is used to generate an array of the points near the collision point. This array determines which points are deformable given the users current contact position. The size of the deformed region is also based on the magnitude of the force the user is exerting on the virtual object.

Once a collision is detected, the force vector to apply to the user is calculated from the distance between the collision point and a haptic probe. We also use the magnitude of this force vector to determine the size of the region of the virtual object to deform. The deformation region is hemispherical in shape, with a radius determined by the magnitude of the force vector. The actual vertices to be deformed are determined using the search algorithm in kd-tree (described above). The deformation force vector is computed based on mass-spring methods assigned to each vertex or connect vertices. If the reaction force vector is increased by the user's action, the region to be deformed is similarly enlarged by the calculated reaction force.

3 Demonstration

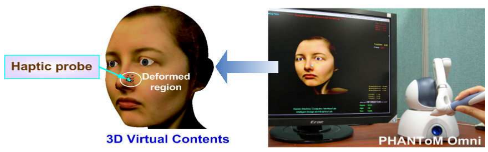

Figure 1: Implementation results.

The proposed algorithm was implemented using a PHANToM Omni haptic device by Sensable technology. Its performance up to 100K triangles is shown to be stable and real-time haptic interaction with 1 KHz haptic rate. Figure 1 shows results of the proposed haptic deformation. A user can create a new content by exploring, manipulating and modifying virtual content with 3D haptic interaction intuitively.

4 Conclusion

This paper proposed a new haptic deformation algorithm in order to allow a user with versatile touch interaction. As the proposed algorithm encourages the spontaneous participation of users through its inclusion of bi-directional haptic interaction, we believe it may enhance user's experience of virtual contents. The proposed algorithm may also be useful as an 3D contents creation and modeling tools for various application areas such as medical, industrial educational, culture domain.

References

J.P. Kim and J. Ryu, "Hardware Based 2.5D Haptic Rendering Algorithm using Localized Occupancy Map Instance", ICAT, pp. 132-137, 2004.

Andrew Moore, "An introductory tutorial on kd-trees", PhD. Thesis, Technical Report No. 209, Computer Laboratory, University of Cambridge, 1991.

*e-mail:bclee@gist.ac.kr
†e-mail:lowtar@gist.ac.kr
‡e-mail:jucpark@gist.ac.kr
§e-mail:lee@kyebek.gist.ac.kr
¶e-mail:ryu@gist.ac.kr