An Efficient Dynamic Point Algorithm for Line-based Collision Detection in Real Time Surgery Simulation Involving Haptics

Anderson MACIEL [1], Suvranu DE
Department of Mechanical, Aerospace & Nuclear Engineering, Rensselaer Polytechnic Institute, Troy/NY

Abstract. In this paper, we introduce a novel "dynamic point" algorithm for computing the interaction of a line-shaped haptic cursor and polygonal surface models which has a near constant complexity. The algorithm is applied in laparoscopic surgery simulation for interaction of surgical instruments with physics-based deformable organ models.

Keywords. Computer Graphics, Interaction techniques, Simulation and Modeling, Haptic I/O

Introduction

In real time computer graphics, "interactivity" is limited to a display rate of 30 frames per second. However, in multimodal virtual environments involving haptic interactions, a much higher update rate of about 1 kHz is necessary to ensure continuous interactions and smooth transitions. The simplest and most efficient interaction paradigm in such environments is to represent the haptic cursor as a point. However, in many situations, such as in real time medical simulations involving the interactions of long slender surgical tools with soft deformable organs, such a paradigm is nonrealistic and at least a line-based interaction is desirable. While line-based methods exist, the main impediment to their widespread use is the associated computational complexity. In this paper, we introduce a novel "dynamic point" algorithm for computing the interaction of a line-shaped haptic cursor and polygonal surface models which has a near constant complexity.

1. Related work

Surgery simulation in virtual reality relies on realistic models of the human body and advanced interaction techniques to manipulate such models. Examples of works on de-

[1]Corresponding Author: Anderson Maciel, MANE-RPI, 110 8th st. Troy, NY 12180, USA; E-mail: maciea@rpi.edu.

formation models for surgery simulation are [1], [2] and [3]. The present work, however, does not focus on the deformation itself. It lies at the intersection of other two important areas of active research in interactive surgery simulation: collision detection and computer haptics. Collision detection involves checking the geometry of the target objects for interpenetration, which is usually made using static interference tests. These tests include distance calculation algorithms [4] [5] and intersection checks. Bounding boxes intersection test is particularly useful in bounding volume methods. They can be tested using the separating axis test [6]. Intersection test between triangles can be done efficiently with the algorithm of [7]. Triangle-triangle test are useful because most 3D polygonal representations are triangle meshes. We refer to [8], [9] and [10] for detailed surveys on collision detection.

Computer haptics, on the other hand, is analogous to computer graphics and deals with various aspects of computation and rendering of force information when the user interacts with virtual objects. For general and psychophysical aspects of haptic interactions and on haptic rendering we refer to [11] and [12].

Here, we review the literature on collision detection for interactive virtual environments, especially those that involve haptics. In these environments, three clearly different contact situations exist: point to model, line to model and geometry to model. Haptic rendering algorithms make use of basic interference tests to provide force-feedback. The first haptic rendering algorithms were based on vector field methods [13]. Then, to overcome the many drawbacks of vector fields, the concept of god-object was introduced in [14]. A god-object is a virtual model of the haptic interface which conforms to the virtual environment. Recently, [15] introduced an extension of the god-object paradigm to haptic interaction between rigid bodies of complex geometry. Although not explicitly used in haptics, the Voronoi-clip (V-clip) algorithm [5] presents the use of spatial-coherence and closest feature idea in collision detection, which is also adopted in our work. Another work exploring local features and spatial and temporal coherence with application to simple haptic environments may be found in [16].

Boundary volumes and spatial partitioning have also been used to detect collisions in haptic applications. A hybrid approach based on uniform grids using a hash table and OBB-trees is presented in [12]. OBB-trees are also used in [17] to reduce the problem to local searches. In contrast to point-based methods, a ray-based haptic rendering algorithm is proposed in [18] that enables the user to touch and feel convex polyhedral objects with the haptic probe shaped as a line segment. They further explored the use of torque-feedback coupling two Phantom devices. A dynamic method to compute point to mesh distances is presented in [19]. Multiresolution hierarchy and bounding volumes are used to achieve constant time queries in some cases.

2. Tools and Methods

In applications involving interactions of a punctual haptic tool with a polygonal mesh, the Lin-Canny closest feature [20] idea can be exploited to compute the distance of the probe to the closest triangle t and iteratively updating t by checking the distance of its neighbors only. The algorithm complexity is near constant even for deformable objects and fast enough for real time haptic applications. However, in order to maintain realism of simulation, most tools used in laparoscopic surgery are better represented by a line

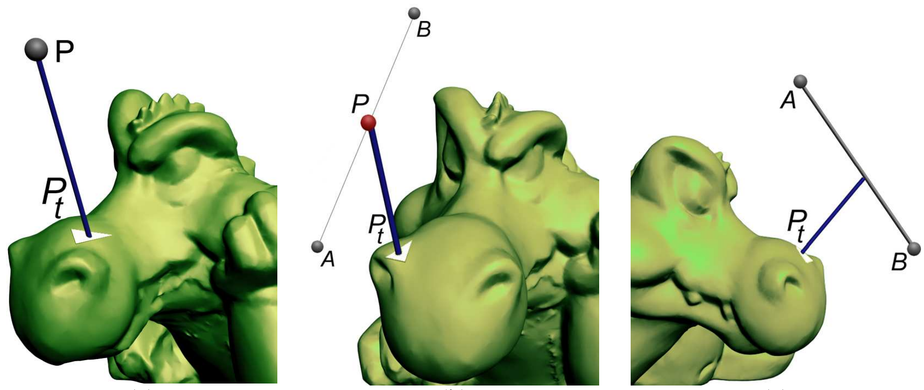


Figure 1. Point-based (a), dynamic point (b) and line-based (c) proximity.

than a point. Such "ray-based" representation is, nevertheless, prohibitively expensive if complex models are used, because the computing time to test collisions increases with the size of the model. To capture the advantages of both point- and ray-based rendering, we propose the "dynamic point" algorithm in which a single point is used for both collision detection and response and is constrained to lie on the line at a location which is instantaneously the closest to the mesh. The data structure is initialized with a full distance check between the triangles and the probe (line) to calculate the nearest point P_t on the mesh to the line (Figure 1b). As the line moves, P_t is updated by checking the distance of only the neighbors of the triangle t originally containing P_t with the line. When the signed distance between the nearest point and the dynamic point is negative a collision is reported. The collision response algorithm is then applied and a penalty force is computed and fed back to the user using the position information of the dynamic point and the closest point.

3. Implementation

We implemented our algorithm on a multi-core PC platform with one graphics card (GeForce 8800). We have developed complete dynamic, textured and shaded physics-based models to highlight that the performance of our collision detection algorithm saves computing power that can be utilized for other important tasks in common applications.

We use the Phantom Omni, by Sensable Inc, a six degrees-of-freedom positional sensing and three degrees-of-freedom force-feedback device to render the forces. We have also developed a plug-and-play interface that allows us to instrument real surgical hand tools and attach them to the Phantom stylus gimbal (Figure 2). Such interface converts the opening and closing of the tool handles into measurable electric potentials using potentiometers [21]. The software interface, in turn, converts the voltage read through a USB interface into angles that define the orientation of instruments movable parts.

We used the interface HDU of the OpenHaptics library to implement a software layer with the haptic device. This layer reads information about the 6-dof cursor from the Phantom into the model and sends back the 3-dof force information. This runs asyn-

Figure 2. The PHANTOM Omni^{TM} haptic interface device with surgical tool interface.

chronously with the collision detection loop, where the dynamic point algorithm is implemented.

The base of our graphics rendering pipeline is OpenGL. However, to obtain improved graphics realism we customized the rendering pipeline using vertex and fragment programs (also called shaders) written in OpenGL Shading Language (GLSL). This allowed us to include effects like textured relief, wetting and multi-layer color texture blending. Examples can be found in figures 3 and 4.

4. Results

As a case-study, we considered a realistic scenario of palpation of internal organs as part of a laparoscopic surgical procedure [22]. We show a partial model of the interior of the abdominal cavity where a few organs are viewed as they would through the laparoscope. The organ models were obtained from the Visible Human project dataset and we used two Phantom Omnis^{TM} as haptic devices. The rigid stomach and spleen have respectively 1863 and 1040 triangles. The liver mesh is composed of 2484 triangles and is simulated by a mass-spring network containing 1312 nodes and 12736 viscoelastic springs. The hook cautery is associated with the line cursor of the right hand for collision detection. The grasper instrument uses three lines for collision purposes, one representing its long shaft and one for each of the claws. Despite the model complexity, the physics loop runs at 550 Hz, the haptics loop at 4500 Hz and the graphics loop at the maximum screen frequency of 60 Hz.

Figure 3 illustrates the case of passing the cautery instrument between the stomach and the liver to lift this latter. Observe that the result of collision is the one of a line contact, even if the dynamic point algorithm considers only one instantaneous point on the line at a time for collision purposes. Another case, this one using two instruments at the same time is presented in Figure 4. Here, a grasper is used to lift part of the liver allowing the cautery-hook to access behind it.

5. Conclusions/Discussion

We have presented the dynamic point algorithm as the most efficient algorithm for line-based collision detection and response surgery simulation involving haptics. It can han-

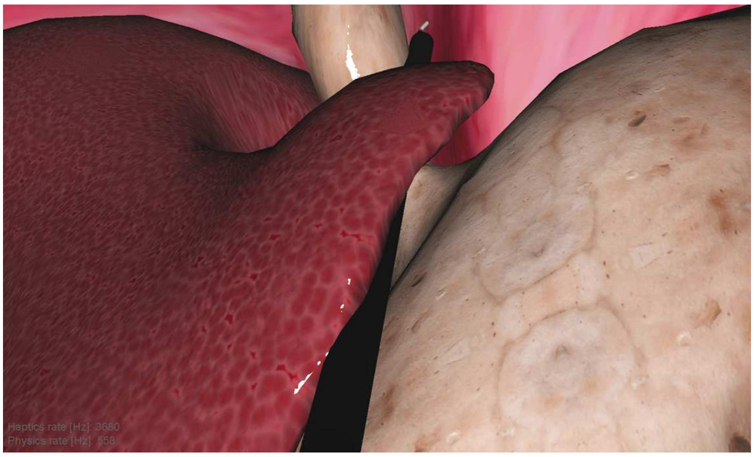
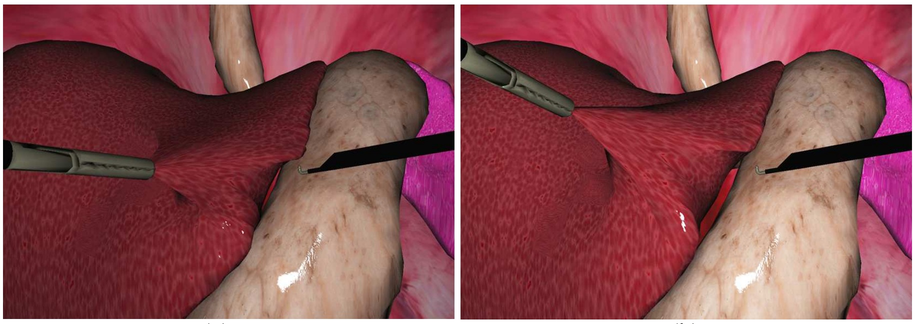


Figure 3. The liver is being lifted by the surgical tool represented by a single dynamic point traversing along its length.



(a) (b)

Figure 4. A bimanual intervention where the liver is lifted by a grasper allowing the cautery-hook to access behind it.

dle very large deformable triangle meshes involving tents of thousands of triangles at haptic frame rates of more than 10 kHz for both rigid and deformable objects. We have implemented and evaluated the algorithm within a complete graphics-haptics-physics-based frame-work. Finally, we presented a case-study on virtual laparoscopic surgery to illustrate the use of the dynamic point in a practical interactive application.

Further research should focus on the ability of these algorithms to provide training and planning mechanisms to surgeons and trainees.

Acknowledgements

This work was supported by grants R21 EB003547-01 and R01 EB005807-01 from NIH/NIBIB.

References

[1] Gábor Székely, Ch. Brechbuehler, R. Hutter, A. Rhomberg, and P. Schmid. Modeling of soft tissue deformation for laparoscopic surgery simulation. In *MICCAI '98: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 550–561, London, UK, 1998. Springer-Verlag.

[2] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):62–73, 1999.

[3] Xunlei Wu, Michael S. Downes, Tolga Goktekin, and Frank Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshe. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3), pages 349–358. Blackwell Publishing, 2001.

[4] A. Joukhadar, A. Wabbi, and C. Laugier. Fast contact localisation between deformable polyhedra in motion. In *Proceedings of Computer Animation '96*, pages 126–135, June 1996.

[5] Brian Mirtich. V-clip: fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17(3):177–208, 1998.

[6] S. Gottschalk. Separating axis theorem. Technical Report TR96-024, UNC Chapel Hill, 1996.

[7] Tomas Möller. A fast triangle-triangle intersection test. *J. Graph. Tools*, 2(2):25–30, 1997.

[8] M. C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. In *Proceedings of the 8th IMA Conference on Mathematics of Surfaces*, pages 37–56, 1998.

[9] P. Jiménez, F. Thomas, and C. Torras. 3D Collision Detection: A Survey. *Computers and Graphics*, 25(2):269–285, Apr 2001.

[10] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. *Computer Graphics forum*, 24(1):61–81, mar 2005.

[11] William R. Mark, Scott C. Randolph, Mark Finch, James M. Van Verth, and Russell M. Taylor II. Adding force feedback to graphics systems: Issues and solutions. In *SIGGRAPH*, pages 447–452, 1996.

[12] A. Gregory, M.C Lin, S. Gottschalk, and R. Taylor. Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device. *Comput. Geom. Theory Appl.*, 15(1-3):69–89, 2000.

[13] T. H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In C. J. Radcliffe, editor, *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, 1994. ASME.

[14] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3*, page 3146, Washington, DC, USA, 1995. IEEE Computer Society.

[15] Michael Ortega, Stephane Redon, and Sabine Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies. In *VR '06: Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, page 27, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Young J. Kim, Miguel A. Otaduy, Ming C. Lin, and Dinesh Manocha. Six-degree-of-freedom haptic rendering using incremental and localized computations. *Presence: Teleoper. Virtual Environ.*, 12(3):277–295, 2003.

[17] Chih-Hao Ho, Cagatay Basdogan, and Mandayam A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence*, 8(5):477–491, 1999.

[18] Chih-Hao Ho, Cagatay Basdogan, and Mandayam A. Srinivasan. Ray-based haptic rendering: Force and torque interactions between a line probe and 3d objects in virtual environments. *I. J. Robotic Res.*, 19(7):668–683, 2000.

[19] André Guéziec. 'meshsweeper': Dynamic point-to-polygonal-mesh distance and applications. *IEEE Trans. Vis. Comput. Graph.*, 7(1):47–61, 2001.

[20] Ming C. Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, EECS Department, University of California, Berkeley, 1994.

[21] Suleiman BaniHani, Anderson Maciel, and Suvranu De. Intuitive surgical interface for laparoscopic surgery. In *Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*. IEEE, 2008. submitted.

[22] Cagatay Basdogan, Suvranu De, Jung Kim, Manivannan Muniyandi, Hyun Kim, and Mandayam A. Srinivasan. Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64, 2004.