Accelerated Haptic Rendering of Polygonal Models through Local Descent

David E. Johnson and Peter Willemsen
School of Computing,
University of Utah

Abstract

In a prior paper, we described a haptic rendering algorithm for arbitrary polygonal models using a six degree-of-freedom haptic interface. In that paper, global search for local minimum distances provided repulsive forces between models. This paper augments that system with a local tracking algorithm that quickly updates and maintains globally computed minima. The global search continuously adds and deletes local minimum distance pairs being updated by the local search. This new system supports higher haptic interaction rates on more complex scenes than the previous approach.

1 Introduction

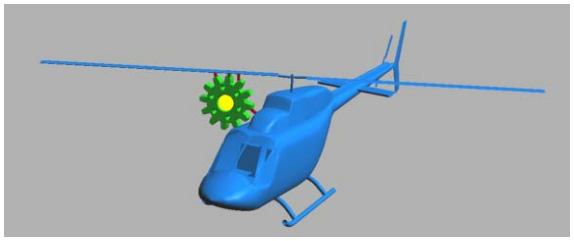

Figure 1: *Our system can haptically render models like a 6,000 triangle gear model and a 113,000 triangle helicopter model.*

Haptic interfaces allow people to naturally manipulate virtual models with their sense of touch. Sensations of contact between models are created by simulating interactions in the virtual environment and then reflecting the interaction forces to the user through a robotic interface. Since haptic interfaces require very fast update rates, a computational bottleneck in this process is determining interaction between models in the virtual environment.

In a prior paper[9], we computed *local minimum distances* (LMDs) between polygonal models using *spatialized normal cone hierarchies* (SNCH)[8]. The SNCH data structure hierarchically encapsulates the position and spread of surface normals over a model. By hierarchically pruning away portions of two models that cannot point towards each other, we can efficiently find local minimum distances between two polygonal models. These minima provide the basis for spring based repelling forces that prevent models from interpenetrating.

In this paper, we develop a local gradient search for maintaining LMDs between models during model movement. This local gradient search is initialized by the global spatialized normal cone hierarchy search for LMDs and periodically updated to introduce new LMDs and delete ones that are no longer valid. This multi-stage approach provides much faster computation of the LMDs and allows haptic rendering of environments with much more complicated models.

The haptic rendering algorithm we present is appropriate for activities such as virtual prototyping and adding haptic interaction to virtual environments.

2 Background

Advancements in haptic rendering and geometric computations have been tightly linked. The following sections cover relevant work in distance computations and then review existing work in haptic rendering.

2.1 Distance computations

Almost all the literature on minimum distance, especially for polygonal models, treats the problem primarily as a Euclidean one. Approaches typically partition the model with hierarchical spatial bounding volumes, using primitives such as spheres[17], convex polytopes[16], oriented swept sphere volumes[11], and convex surface patches[3]. Nodes of the hierarchy are pruned by comparing lower bound distances between nodes to upper bounds on the global distance. This approach returns the global minimum distance between models.

A different approach is found in work on sculptured surfaces, such as B-splines[13][15]. These approaches develop techniques for tracking the locally closest or extremal points between two surfaces after initialization by a global search[7]. The set of equations that describe these local distances is based on collinearity of normals rather than Euclidean distance.

In [8], the collinearity approach is adapted to polygonal models through the spatialized normal cone hierarchy data structure. This technique efficiently finds all local minima between polygonal models.

2.2 Haptic rendering

Haptic rendering algorithms were first developed to support three DOF haptic interfaces. These algorithms and devices support a moving point touching a computer model. Researchers have developed algorithms for haptic point contact with polygonal[1] [4][18][19][6], implicit[20], and NURBS[22][22] models.

More recently, efforts have focused on developing techniques to haptically render the interactions between two models. The resulting forces include torques as well as translation forces, and a six DOF haptic device is needed to accurately reflect the results back to a user.

In the polygonal model domain, the first efforts at six DOF haptic rendering were for small convex shapes[2]. More recently, research has looked at collections of convex bodies[5], as well as incremental methods for computing the penetration depth[10]. Most recently, the convex decomposition approach has been extended with perceptual level of detailing to accelerate haptic rendering for very large models[14].

Model-model haptic rendering for general NURBS models is developed in [13]. This system uses a three pass approach: initial distance monitoring using polygonal approximations, local closest point initialization using Newton's method on an extremal distance formulation, and stable maintenance of the penetration depth distance with a velocity relation between parametric space and Euclidean movement. Another method based on control theory maintains the extremal distance even with imprecise starting values [15].

Researchers at Boeing have taken a different approach than penetration-based methods by creating a voxel-based scene and allowing a point-sampled free-moving model to interact with the voxels[12]. The advantage is that the computation time can be tightly bound by the number of voxels and the number of points in the free-moving model. They also created a voxel boundary around the models in the scene to prevent interpenetration of models, which would invalidate the correctness of the virtual prototyping.

In [9], we adapted the use of SNCHs to model-model haptic rendering by using the computed local minimum distances as repelling springs and by reducing the scope of the search. This follows the Boeing paradigm of not allowing interpenetration of models. However, in this system, the models in the scene are more easily inserted or moved than with a voxel-based approach.

3 Approach

This paper adds a local search to speed intermediate time steps between global updates. Our haptic rendering algorithm first computes all LMDs within a cutoff distance using the global SNCH search. These LMDs are fed to the local update thread, which performs local gradient descent on the LMDs given new positions of the models. The updated LMDs are used to compute forces and torques repelling the models.

The local update works as fast as it can on the LMDs it knows about. Concurrently, the global search computes new LMDs. When it finishes a time step, it notifies the local search that new LMDs are available. The local search then updates these new LMDs to the current model positions and continues local updates.

3.1 Local Search

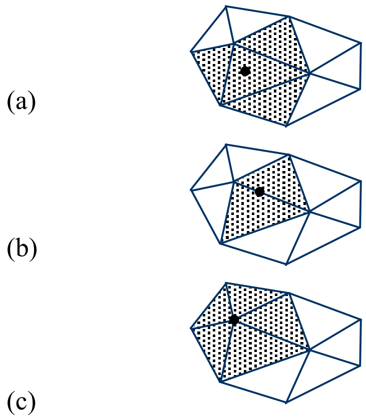

Figure 2: *The three possible neighborhoods to be checked depend on the minimum distance point from the last time step. In (a), the minimum distance point was on a triangle face, and it and all its adjoining triangles form the neighborhood. When the point from the previous step was on an edge, as in (b), the triangles that share that edge form the neighborhood. The final case (c) is for a vertex, where all triangles that share that vertex form the neighborhood.*

A pair of points, one on each model, forms each LMD. After a model moves, the local search algorithm looks at the neighborhood around each LMD point and computes the distances between all the triangles in one neighborhood and all the triangles in the other model's neighborhood. If any of these triangle pairs are closer than the current LMD, then the search continues with those triangles' neighborhoods until the minimum distance converges. The points that form this new minimum distance are the updated LMD.

If the last LMD point was on the face of a triangle, then the local neighborhood is defined to be the triangle plus the three triangles bordering its edges. If the last point was on an edge, only the two triangles that share that edge are part of the local neighborhood. When the last point was at a vertex, all triangles that share that vertex are searched for a new LMD.

3.2 Computational Efficiency

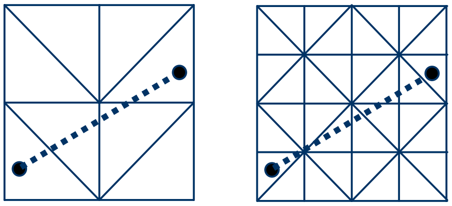

Figure 3: *The number of triangles crossed grows roughly by the square root of the number of triangles in the model.*

The number of neighborhoods that must be checked varies with the model resolution and the movement of the models. For models with reasonably formed triangles, the number of triangles searched on one model grows roughly as the $\sqrt{n}$, where n is the number of triangles. Figure 3 shows the resolution quadrupling and the number of triangles crossed growing by a little more than a factor of two. In addition, for haptic rendering running near 1000 Hz, temporal coherence is very high and the number of triangles crossed is small.

The global search efficiency is dependant on the number of LMDs as well as the complexity of the two models. In the best case, the two models are of balanced complexity and there is a single LMD, and the global search takes *log(n)* time, where n is the number of triangles in a model. In a more typical scenario, there are multiple LMDs and the pruning cannot always remove nodes that will not eventually yield minima, increasing search time. In cases of manifold minima solutions, such as two parallel planes, the search takes *nlog(n)* time. However, manifold solutions are rarely encountered with human-guided model interactions. Additionally, the global search is improved with a cutoff distance[9], so only close portions of the models are searched for minima. This reduces the apparent complexity of the model and the number of LMDs the local update must process, especially with bumpy models.

3.3 Forces and Torques

At each time step in the haptic rendering loop, our algorithm computes the LMDs that are closer than a cutoff distance between the model that is controlled by the haptic interface and the rest of the models in the scene. Each LMD is considered a virtual spring with a rest length equal to the cutoff distance. Each spring is attached to the models by the pairs of points that form the LMD. See [9] for a more detailed treatment of the force and torque computation.

3.4 Preprocessing

The LMD computations require precomputing a spatialized normal cone hierarchy for each polygonal model in the virtual prototyping environment. However, models in the scene can be added and deleted, or moved around interactively, without needing further precomputation. The preprocessing step takes a few seconds for models of a few thousand triangles.

The local search routine uses local topological connectivity between the triangles of the models. Unstructured triangle list models go through a one-time conversion from triangle list data structures to vertex-edge-triangle lists with neighbor information.

3.5 System overview

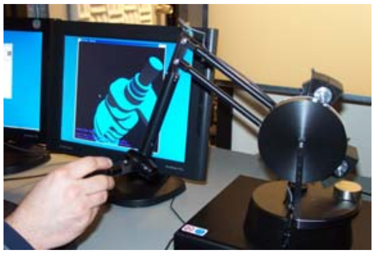

Figure 4: *The 6DOF Phantom.*

Our virtual prototyping system is based on a Sensable six DOF PHaNTOM haptic interface (Figure 4). The computations run on a dual processor Pentium 4 2.4 GHz Linux computer with a gigabyte of RAM and a GeForce 4 Ti 4400 graphics card. This is the same system used in our prior paper[9], making performance comparisons more accurate.

3.6 Multi-threading

This type of application would be difficult to write as a single thread of computation. The application uses three threads: a global search thread, a local update thread, and a graphics thread. This architecture allows us to restrict the computational load of the graphics and global threads, and let the local update run as fast as possible. On a two-processor system, this translates into the local update getting one processor to itself and the other threads sharing the other processor.

4 Results

The local search algorithm computes updated forces and torques at kilohertz rates. When model complexity grows, the global search tends to slow down, but the local update speed is mostly dependant on the number of LMDs, not the complexity of the model. The global search is still the limiting factor in environment complexity.

We instrumented the local update thread to record the time to compute the local update, the number of triangle pairs searched during the local update, and the time for the global search to compute the LMDs. The following figures show these results for a variety of model-model interactions. For all these examples, the top graph represents the local update time, the middle graph the number of triangle-pairs searched during the local update, and the bottom graph the time for the global LMD computation to update.

4.1 Crankshaft-Gear

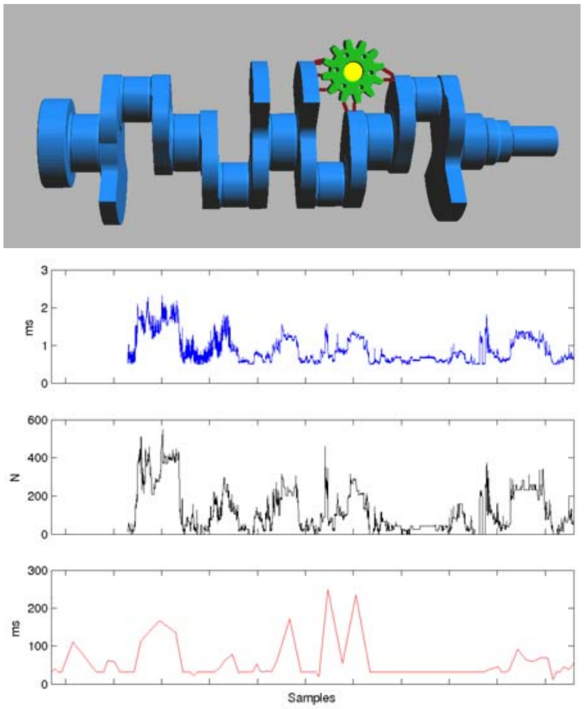

Figure 5: *The crankshaft model has 45,000 triangles and the gear has 6,300 triangles.*

The top graph shows the time the local update took to update the LMDs and compute forces. The local update was able to maintain near kilohertz rates even during complex interactions. The middle graph counts the number of triangle pairs searched during each local update. The local update time correlates well with the number of triangles searched. The bottom graph shows the computation time for the global search to find the LMDs. Without the local update, haptic interaction would have been highly unstable and slow.

4.2 Spring-Teapot

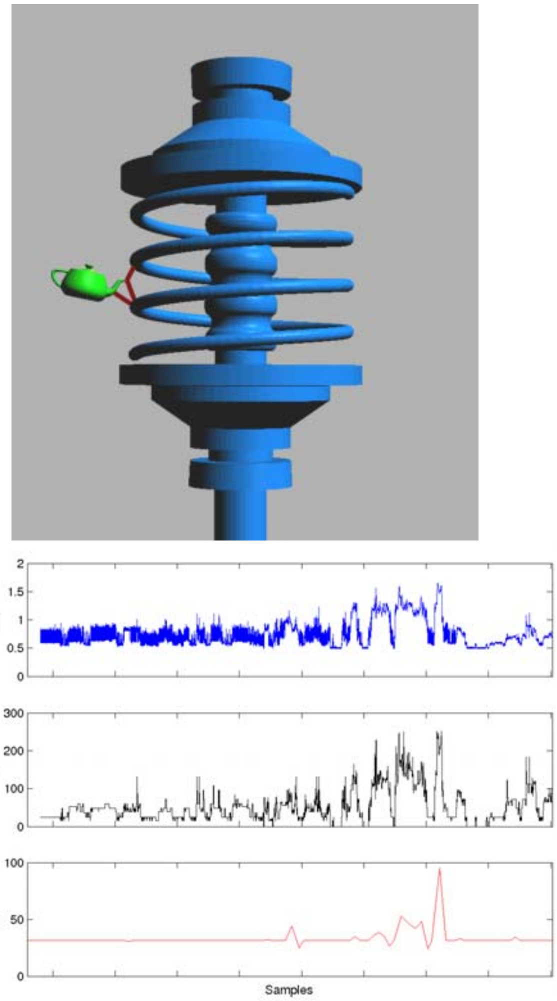

Figure 6: *The spring model has 23,500 triangles and the teapot has 5,600 triangles.*

In this example, we moved the teapot around the spring part and explored the coils of the springs. Even though the models are high resolution, typically there were only a few LMDs to track, and the local update was able to maintain a high update rate.

4.3 Horse-Bunny

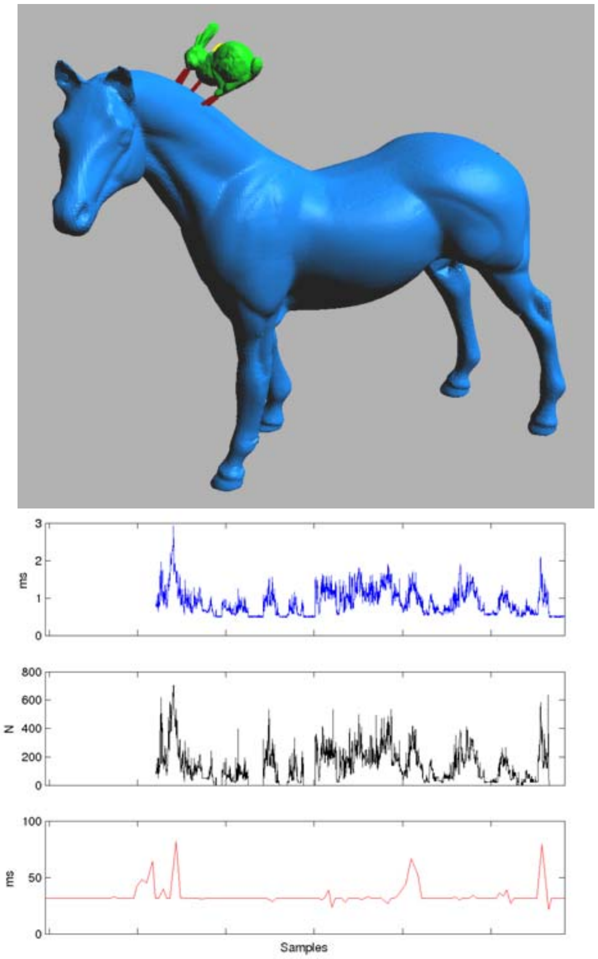

Figure 7: *The horse model has 97,000 triangles and the bunny model has 2,200 triangles.*

Non-mechanical models, such as the horse and bunny, provide additional challenges to our haptic rendering system, as the finely detailed surfaces can produce nearly redundant local minima. In this example, we are still able to update the LMDs and forces at around 1 kilohertz.

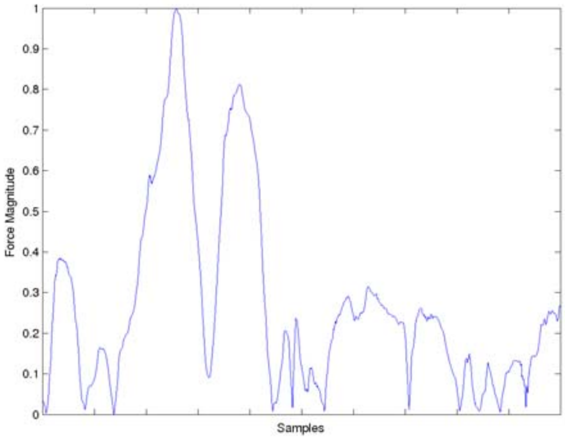

Figure 8: *This graph shows the magnitude of the translation forces during the horse-bunny interactions.*

The forces of interaction feel smooth. Figure 8 shows the magnitude of the translational forces during haptic interaction between the horse model and the bunny. The large-scale bumps are from moving the bunny model around the horse and bumping against it. Smoother responses are possible when continuously pressing the two models together. However, the lack of high-frequency fluctuations shows good haptic rendering stability. Smooth force response is possible because forces are derived only from the LMDs in the fast local update thread and because new forces are typically introduced near the cutoff distance, where the force for that LMD is zero.

5 Conclusion

This work dramatically improves the performance of our previous haptic rendering system. We are able to interact with models that are an order of magnitude larger at update rates an order of magnitude faster than with just using global search for LMDs. The addition of local LMD updates makes the force computation much less dependent on the configuration of the two models. The resulting algorithm is a high performance haptic rendering system suitable for augmenting virtual reality interactions.

6 Acknowledgements

The authors would like to acknowledge support in part from the following grants: NSF DMI9978603, NSF CDA-96-23614, and ARO DAAD 19-01-1-0013.

7 References

[1] Adachi, Y. and Kumano, T. and Ogino, K., “Intermediate Representation For Stiff Virtual Objects,” in *Proc. Virtual Reality Annual Intl. Symp.*, Research Triangle Park, N.C., pp. 203-210, March 11-15, 1995.

[2] Baraff, D., “Fast contact force computations for non-penetrating rigid bodies”, in *Computer Graphics Proceedings, Annual Conference Series*, vol. 28, 1994, pp. 23-34.

[3] Ehmann, S. and Lin, M., “Accurate and Fast Proximity Queries between Polyhedra Using Surface Decomposition”, in Computer Graphics Forum (Proc. of Eurographics), 2001.

[4] Gregory, A., Lin, M., Gottschalk, S., and Taylor, R., “H-COLLIDE: A Framework for Fast and Accurate Collision Detection for Haptic Interaction”, in *Proc. IEEE Virtual Reality '99*

[5] Gregory, A., Mascarenhas, A., Ehmann, S., Lin, M., and Manocha., “Six Degree-of-Freedom Haptic Display of Polygonal Models”, in Proc. IEEE Visualization, 2000.

[6] Ho, C., Basdogan, C., Srinivasan, M.A., “An Efficient Haptic Rendering Technique for Displaying 3D Polyhedral Objects and Their Surface Details in Virtual Environments”, October’99 Vol. 8, No. 5, pp. 477-491, Presence: Teleoperators and Virtual Environments

[7] Johnson, D. E. and Cohen, E., "A framework for efficient minimum distance computations," *Proc. IEEE Intl. Conf. Robotics & Automation*, Leuven, Belgium, May 16-21, 1998, pp. 3678-3684.

[8] Johnson, D. E. and Cohen, E., "Spatialized Normal Cone Hierarchies," in 2001 ACM Symposium on Interactive 3D Graphics, ACM SIGGRAPH, March 2001.

[9] Johnson, D. and Willemsen, P., "Six Degree-of-Freedom Haptic Rendering of Complex Polygonal Models," in Proc. *2003 Haptics Symposium* , 2003.

[10] Kim, Y., Otaduy, M., Lin, M. and Manocha, D. “Six Degree-of Freedom Haptic Display Using Localized Contact Computations”, in the *Tenth Symposium on Haptic Interfaces For Virtual Environment and Teleoperator Systems*, March 24-25, 2002.

[11] Larsen, E., Gottschalk, S., Lin, M., and Manocha, D. “Fast Distance Queries using Rectangular Swept Sphere Volumes”, in Proc. IEEE International Conference on Robotics and Automation , 2000.

[12] McNeely, W., Puterbaugh, K., and J. Troy. “Six degree-of-freedom haptic rendering using voxel sampling”. *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.

[13] Nelson, D., Johnson, D., and Cohen, E., “Haptic Rendering of Surface-to-Surface Sculpted Model Interaction,” in *Proc. 8th Annual Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, (Nashville, TN), ASME, November 1999.

[14] Otaduy, Miguel A., and Lin, Ming C.,”Sensation Preserving Simplification for Haptic Rendering,” Proceedings of ACM SIGGRAPH 2003, San Diego, CA. 2003.

[15] Patoglu, V. and Gillespie, B., “Extremal Distance Maintenance for Parametric Curves and Surfaces,” *IEEE Conference on Robotics and Automation*, Washington D.C., May 11-15, 2002.

[16] Ponamgi, M., Manocha, D., and Lin, M., “Incremental algorithms for collision detection between solid models”, *Proceedings of ACM/SIGGRAPH Symposium on Solid Modeling*, pp. 293-304, 1995.

[17] Quinlan, Sean. "Efficient Distance Computation between Non-Convex Objects," *IEEE Int. Conference on Robotics and Automation*, pp. 3324-3329, 1994.

[18] Ruspini, D., Kolarov, K., and Khatib, O., “The Haptic Display of Complex Graphical Environments,” *in Computer Graphics Proceedings, SIGGRAPH 1997*. Aug. 3-8. pp. 345-352.

[19] Salisbury, K., Brock, D., Massie, T., Swarup, N., and Zilles, C., “Haptic rendering: programming touch interaction with virtual objects”, in *Symp. on Interactive 3D Graphics*, 1995. pp. 123-130.

[20] Salisbury, K., and Tarr, C, “Haptic rendering of surfaces defined by implicit functions”, in *Proc. ASME Dynamic Systems and Control Division*, DSC-Vol. 61, 1997. pp. 61-67.

[21] Sander, P., Gu, X., Gortler, S., Hoppe, H., Snyder, J., “Silhouette Clipping.” in *Computer Graphics Proceedings, SIGGRAPH 2000.* New Orleans. pp. 327-334.

[22] Stewart, P., et al, “CAD Data Representations For Haptic Virtual Prototyping”, Proceedings of DETC’97, *1997 ASME Design Engineering Technical Conferences*, Sept. 14-17, 1997, Sacramento, California.

[25] Thompson II, T.V., Johnson, D.E., and Cohen, E.C. “Direct Haptic Rendering Of Sculptured Models”, in *Proc. 1997 Symposium on Interactive 3D Graphics*, (Providence, RI), April 1997, pp. 167-176.