

## DETC2004/DAC-57507

### A HAPTIC SYSTEM FOR VIRTUAL PROTOTYPING OF POLYGONAL MODELS

**David E. Johnson**  
School of Computing  
University of Utah  
Salt Lake City, USA  
dejohnso@cs.utah.edu

**Peter Willemsen**  
School of Computing  
University of Utah  
Salt Lake City, USA  
willemsn@cs.utah.edu

**Elaine Cohen**  
School of Computing  
University of Utah  
Salt Lake City, USA  
cohen@cs.utah.edu

#### ABSTRACT

Virtual prototyping attempts to replace physical models with virtual models for the purpose of design evaluation. One task a virtual prototyping environment can address is the accessibility of the components of a mechanical system. In this paper, we demonstrate a haptics-based virtual prototyping system for finding collision-free paths for moving models in complex polygonal environments. The system can handle models and environments with hundreds of thousands of triangles, and augments innate human talents at searching for collision-free paths.

#### INTRODUCTION

While mechanical model design increasingly relies upon computer-aided design (CAD) and sophisticated simulation programs, physical prototypes still play an important role in design evaluation. Since physical prototypes are expensive to build, and may take significant time to manufacture, virtual prototyping environments attempt to replace as much functionality of the physical prototypes as possible with a virtual prototype.

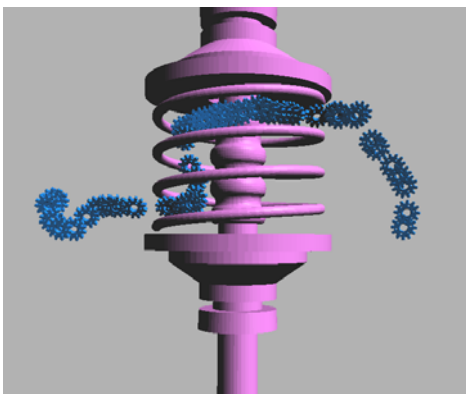


Figure 1: GUIDING A GEAR PAST A SPRING PART IN A SAMPLE VIRTUAL PROTOTYPING SESSION

Accessibility is a design evaluation task that is difficult to simulate on a computer. Two main reasons preclude easy automatic simulation:

- Computation of a collision-free path for complex models is difficult and time-consuming,
- Modeling human manipulation capabilities is difficult.

We propose a haptic system for virtual prototyping that allows human guidance and intuition in developing a collision-free path between virtual models (Figure 1). This type of system provides the intuitive usability of a physical prototype, yet retains the cost and time advantages of a computer model.

Our system has several advantages over existing haptic systems:

- Exact distances between models are computed, so the simulation is accurate to the resolution of the models.
- Haptic feedback is provided before models collide, so no invalid interpenetration of the models needs to occur.
- Models in the environment are freely movable, providing real-time adjustment to the scene as desired.

The virtual prototyping environment is demonstrated on an example mechanical system with hundreds of thousands of triangles.

#### BACKGROUND

Virtual prototyping of accessibility tasks is closely related to the area of path planning. The main distinction is that in virtual prototyping, there is some assumption of human involvement, whereas path planning is usually more of an automatic technique.

Path planning methods fall into two main categories – global methods and local methods. Global methods try to sample the configuration space of the model and the environment, and then connect together collision-free instances into a collision-free path[1][2][3]. Local methods use local

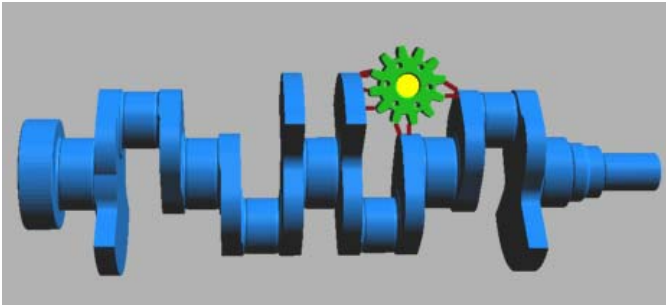


Figure 2: THE SET OF LMDS PRODUCES REPULSIVE FORCES THAT KEEP MODELS FROM COLLIDING.

repulsion techniques to avoid collisions, while being drawn towards a distant goal[4]. However, these local methods can get stuck in local minima and never reach the goal. Our haptics system is similar to the local path planning approach, but uses human guidance to push models past local minima.

Haptics[5] has been proposed as a virtual prototyping interface in prior work. Hollerbach et al. [6] computed fast penetration depths between a point and a spline model to create a sensation of contact with the model, as did researchers at Ford Motor company in [7]. Nelson[8] developed a fast technique for haptic rendering of two spline models in contact and also adapted the method to moving linkages[9]. Ruspini created a haptic rendering system for point interactions with complex, moving polygonal models[10].

McNeely used a six degree-of-freedom (DOF) device to manipulate a point-sampled model with a large-scale voxel environment[11]. The environment in that system is static; however, that approach guarantees a worst case computation time, important for reliable haptic rendering. They report that they were able to use haptics to find collision-free paths in complex environments for which global path planning algorithms failed.

## APPROACH

Our virtual prototyping system is based on a 6-DOF haptic rendering method for complex polygonal models[12][13]. That haptic rendering method uses techniques for finding local minimum distances (LMDs) between polygonal models and computing local updates to the LMDs while the models move. This paper uses those techniques in conjunction with methods for maintaining a collision-free path and visualizing the result to produce a virtual prototyping environment.

The basic haptic rendering approach is to find LMDs between the moving polygonal model and the environment. Each LMD acts like a spring guiding the moving model away from collision with the environment (Figure 2), while human interaction guides the model towards its goal. In order to maintain haptic rates, the set of LMDs are updated using a local gradient search.

These two steps, a search for LMDs and a local update, are summarized in the next two sections, please refer to the original papers for a more complete description.

### Spatialized Normal Cone Pruning

The haptic rendering searches for LMDs using spatialized normal cones[14], which hierarchically encapsulate the position and spread of surface normals over a model. The algorithm uses surface normal information to find portions of each model that

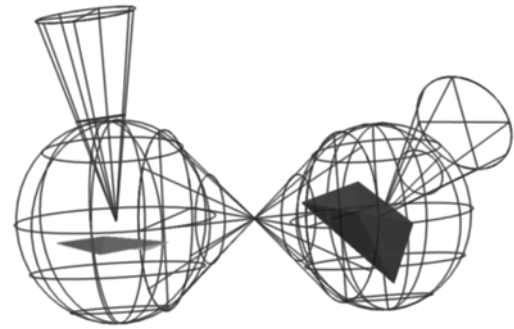


Figure 3: A SPHERE BOUNDS A PORTION OF POLYGONAL GEOMETRY AT A NODE OF THE HIERARCHY. THE NORMAL SPAN OF THE NODE GEOMETRY IS ENCAPSULATED IN A NORMAL CONE. THE RANGE OF POSSIBLE LINES BETWEEN CLOSEST POINTS BETWEEN NODES IS BOUNDED BY A DOUBLE CONE BETWEEN BOUNDING SPHERES. AT EACH PAIR OF NODES, THE ALGORITHM CHECKS TO SEE IF THE NORMAL SPANS OVERLAP AND ARE ALONG A VECTOR CONTAINED WITHIN THE SPAN OF POSSIBLE MINIMUM DISTANCE LINES.

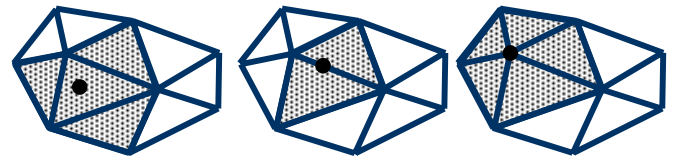


Figure 4: THE LOCAL GRADIENT SEARCH CHECKS NEIGHBORING FEATURES FOR NEW CLOSEST POINTS. WHEN THE LAST POINT IS ON A FACE, NEIGHBORING FACES ARE CHECKED. WHEN THE LAST POINT IS ON AN EDGE, ONLY TWO NEIGHBORING FACES ARE USED. THE LAST CASE IS FOR A VERTEX, WHERE ALL THE TOUCHING FACES ARE CHECKED.

point towards each other and are collinear with the line between potential closest points on the models, a condition which represents a local minimum distance (Figure 3). Nodes in the hierarchy that cannot form a local minimum distance solution are pruned, while remaining nodes are subdivided and the algorithm recursively applied. At leaf triangles, exact tests compute whether a local minimum distance solution exists.

### Local Search with Gradient Descent

Each LMD pair from the normal cone computation represents locally closest points on the two models. The LMD pairs are updated at haptic rates by a local gradient search algorithm[13]. The gradient search checks neighboring features on the polygonal model for a new pair of points that are closer than the current pair (Figure 4). This search repeats for each LMD until the LMD converges to a new distance.

### Combined Search and Update

As fast as the global normal cone search can compute LMDs, it introduces new LMDs and deletes those that are no longer needed. In our examples, this update happened at 10-100Hz. The local gradient search then updated the LMDs at over 1000Hz for stable haptic rendering.

### Force and Torque

Each LMD pair within a cutoff distance contributes to the total force and torque being applied to the model under the control of the force-feedback device. We approximate the center of mass and moments of the moving model using an oriented bounding box approximation. These forces and torques are reflected back to the user by the 6-DOF force-feedback device.

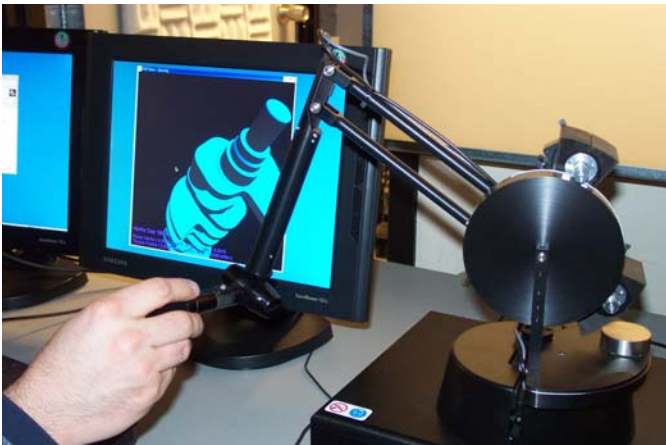


Figure5: THE 6-DOF PHANTOM.

## SYSTEM DESIGN

Our virtual prototyping system is based on a Sensable six DOF PHANTOM haptic interface (Figure 5). 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.

The software architecture uses a multi-threaded design to handle the different rates of computation needed for graphics, global normal cone search, and the local updates with force computations. 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 second processor.

## VIRTUAL PROTOTYPING SYSTEM

Since we compute LMDs while the moving model is still some distance from the environment models, haptic forces are used to guide the moving model away from collision with the environment (Figure 6). The onset distance for forces is adjustable, so the user can decide how much clearance between models is desired during testing. In general, the LMDs tend to approximate the local distance field between the models, and the forces tend to push the moving model towards the medial axis between the models. Since the medial axis is the surface of maximum clearance between models, these forces tend to guide the moving model towards the safest path.

## Collision-Free Path

While the test object is being moved by the haptic interface, its position and orientation are stored in a buffer. This buffer allows the motion of the test object to be played back for review, analysis, or further modification.

If the moving model is forced to penetrate an environment model by the user, the simulation is no longer valid. A collision state is detected and the simulation is rolled back, using the stored positions and orientations in the buffer, until the model state is valid. The simulation can then resume, and the user can try new approaches for finding a collision-free path. This means that the path stored by our virtual prototyping program

is always valid, and if the moving model can reach its goal, the problem has been solved.

## Detecting Collisions

Collisions are detected when the smallest LMD falls below an adjustable parameter. This parameter can represent error in the fit of the polygonal model to an original CAD model, or it can represent a desired minimum clearance between models. Detecting collisions in this fashion, instead of with actual model intersection, provides more control over simulation accuracy.

## Path Visualization

Since we store model positions and orientations during the simulation, a sampling of the path of the model can be visualized. Drawing a copy of the moving model at each sampled location (or some subset), allows the user to check the validity of the collision-free path, and to examine any unusual maneuvering needed to safely guide the model. One drawback is that if many positions are visualized simultaneously, the frame rate of the display can slow.

## Interface

The main interface is the 6-DOF haptic interface. After loading models into the environment, the position of the currently selected model is controlled by the user moving the haptic interface. The selected model is changed with keyboard commands, so any model in the environment is freely movable by the haptic interface.

Keyboard commands also control the recording of the collision-free path, stopping of recording, and visualization of the path in playback mode.

The current set of LMDs are displayed as red lines between the two models. They help provide feedback cues to the relative positions of the two models in the absence of stereo viewing.

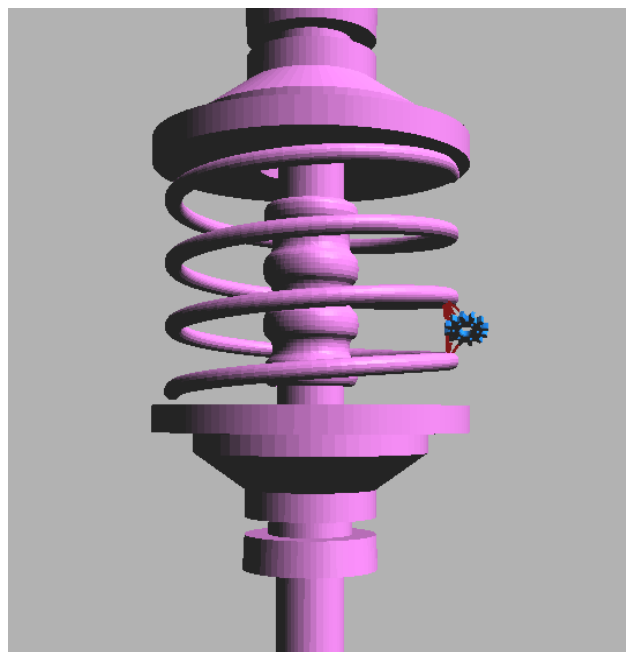


Figure 6: THE LMDS PROVIDE GUIDANCE IN REGIONS OF LIMITED CLEARANCE.

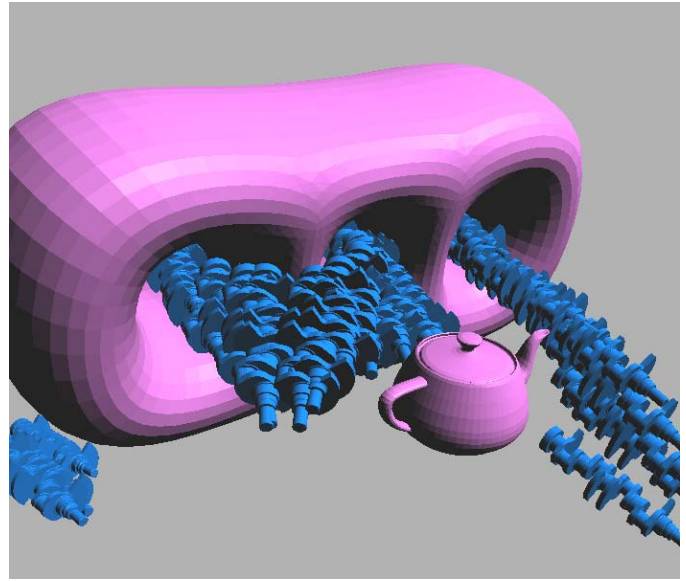
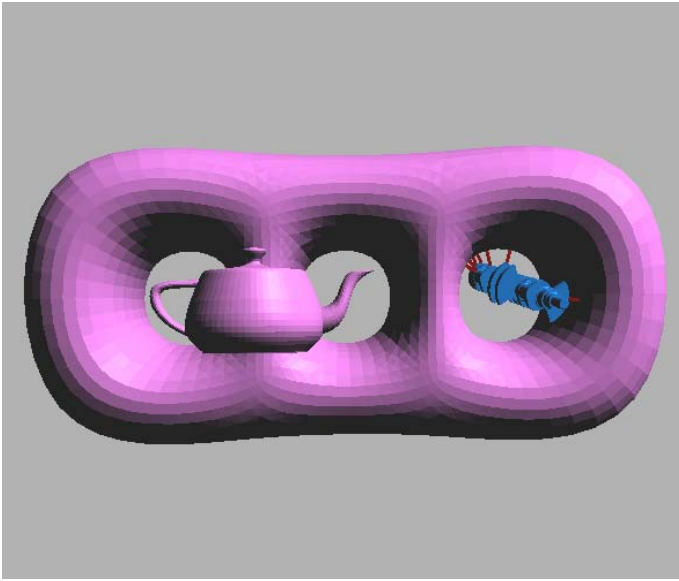


Figure 7: HAPTICS GUIDES THE CRANK MODEL THROUGH THE HOLES WHILE AVOIDING THE TEAPOT MODEL. THE FINAL PATH IS VISUALIZED AS A SAMPLING OF MODEL POSITIONS DURING THE SIMULATION.

## RESULTS

We tested our virtual prototyping system with a variety of models. In the tests, we threaded one moving model around and inside the environment model. The tests were such that the moving models needed to be oriented properly to fit through the gaps in the environment model, so that success without haptic feedback would be difficult. The model and environment sizes ranged from around 6,000 triangles to 113,000 triangles. In all the cases, we were able to intuitively find a collision-free path to accomplish the goal.

### Gear-Spring Part

In the first test, we used a gear model with 6,300 triangles and a spring part model with 23,500 triangles. The goal was to have the gear enter the spring, traverse down the body, and then exit the spring. There was limited clearance between the gear and the spring and spring body, as well as between the coils of the spring. The gear model had to be turned almost flat to fit through the coils. However, with haptics, the forces naturally guided the model (Figure 6) along a safe path (Figure 1).

### Crank-Holes-Teapot

In this example, we used a crank model with 45,000 triangles, a three-holed block with almost 12,000 triangles, and a teapot model with 5,600 triangles. We used the haptic interface to position the block and the teapot in such a way that there was not a clear path from one hole to the next. The goal in this test was to thread the crank model through all three holes while avoiding the teapot.

The haptic interface provided enough cues to the user to find a path out of the middle hole and to tilt around the teapot, even though that portion of the path was occluded by the teapot during the test (Figure 7 shows the view during the test and a tilted view after the test for the path visualization).

### Helicopter-Rocker

The final test used a helicopter model with 113,000 triangles and a rocker arm model with 40,000 triangles. For this test, we wanted to pass the rocker through the open window of the helicopter, around the interior, and back out, avoiding other structures such as the helicopter blade and tail (Figure 8).

The haptic rendering system was able to provide useful feedback during this test, creating a collision-free path under user guidance.

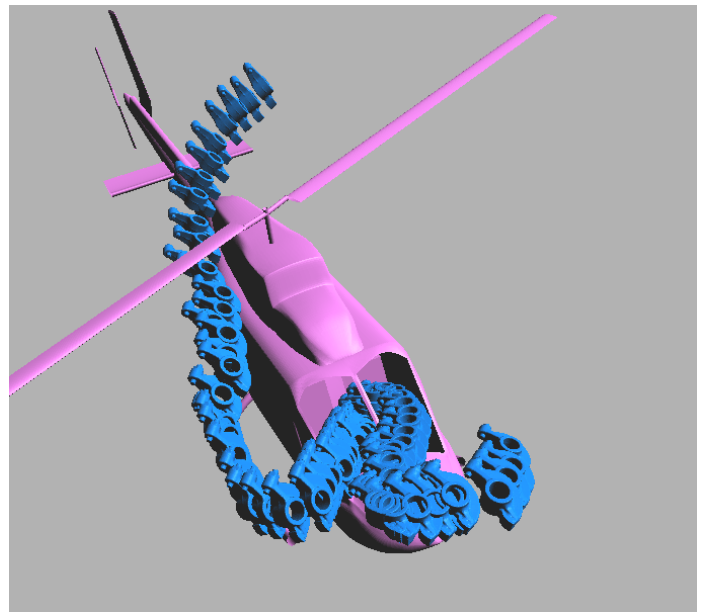


Figure 8: THIS TEST INVOLVED MODELS WITH OVER 150,000 COMBINED TRIANGLES.

## DISCUSSION AND CONCLUSION

The presented system advances the state-of-the-art in haptically-enhanced virtual prototyping systems by allowing virtual prototyping on general, freely positioned, polygonal models. In addition, our mechanism for rolling back model collision states to a collision-free position and orientation simplifies motion planning by always storing a safe path. The use of adjustable distances for both force onset and collision distance also enhances the capabilities of the system by simulating different clearance constraints during the task.

Comparison with other systems is difficult. Most other haptic rendering methods for polygonal models depend on penetration to generate forces, which would invalidate the simulation results. The highest-performing system[15] to compare with can render similarly sized models as our system but uses potentially low resolution levels of details in computing forces. In contrast, we are able to compute exact distances between high-resolution models and generate forces before penetration.

The tests demonstrated the system on a variety of model types and sizes. While the chosen models were not solving an actual real-world problem, the model shapes, resolutions, and task types are representative of the kinds of problems the system can solve.

The locality of the LMD computation means that these environments can scale to very large number of triangles, since most of the triangles will not come into consideration at any one time.

We hope to improve the power of the virtual prototyping system by continuing to improve the speed of the haptic rendering sub-system, by including other measures besides distance as a way of generating forces (such as clearance), and giving feedback to the user indicating the quality of the path as it is being generated.

## ACKNOWLEDGMENTS

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.

## REFERENCES

- [1] J.F. Canny. The Complexity of Robot Motion Planning. ACM Doctoral Dissertation Award. MIT Press, 1988.
- [2] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, pages 12(4):566–580, 1996.
- [3] M. Foskey, M. Garber, M. Lin, and D. Manocha, "A Voronoi-Based Hybrid Motion Planner", Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems, 2001.
- [4] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots," in *IJRR*, 5(1):90–98, 1986.
- [5] K. Salisbury et al., "Haptic rendering: programming touch interaction with virtual objects", in *Symp. on Interactive 3D Graphics*, 1995. pp. 123-130.
- [6] J. Hollerbach et al., "Haptic interfacing for virtual prototyping of mechanical CAD designs," ASME Design for Manufacturing Symposium, (Sacramento, CA), Sept. 14-17, 1997.
- [7] P. Stewart 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.
- [8] D. Nelson, D. Johnson, and E. Cohen, "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.
- [9] D. Nelson and E. Cohen., "Optimization-Based Virtual Surface Contact Manipulation at Force Control Rates," in IEEE Virtual Reality 2000, (New Brunswick, NJ), IEEE, March 2000.
- [10] D. Ruspini, K. Kolarov, and O. Khatib, "The Haptic Display of Complex Graphical Environments," in *Computer Graphics Proceedings, SIGGRAPH 1997*. Aug. 3-8. pp. 345-352.
- [11] W. McNeely, K. Puterbaugh, and J. Troy. "Six degree-of-freedom haptic rendering using voxel sampling". *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.
- [12] D. Johnson and P. Willemsen, "Six Degree-of-Freedom Haptic Rendering of Complex Polygonal Models," in Proc. *2003 Haptics Symposium*, 2003.
- [13] D. Johnson and P. Willemsen, "Accelerated Haptic Rendering of Polygonal Models through Local Descent," in Haptics Symposium 2004, IEEE, 2004.
- [14] D. Johnson and E. Cohen, "Spatialized Normal Cone Hierarchies," in Proc. *2001 ACM Symposium on Interactive 3D Graphics*, Research Triangle Park, NC, March 19-21, 2001. pp. 129-134.
- [15] M. Otaduy and M. Lin, "Sensation Preserving Simplification for Haptic Rendering", in Proceedings of ACM SIGGRAPH 2003/ACM Transactions on Graphics, Vol. 22, pp. 543-553, San Diego, CA. 2003.