Building and Using Polyhedral Hierarchies *

David Dobkin Ayellet Tal
Department of Computer Science
Princeton University
Princeton, NJ 08544

This animation shows the techniques used to build a hierarchy of polyhedra as a preprocessing step for intersection detection algorithms. It is an animation of ideas developed in the papers [1,2]. We show here how the hierarchy is created from a polyhedron and animate an algorithm for detecting the intersection between a polyhedron and a plane.

The hierarchical data structure gives a concise representation for polyhedra that is useful in various searching tasks. Given a polyhedron, P, we build a hierarchy $P_0 = P, P_1, ..., P_k$ such that P_k is a tetrahedron. Each P_{i+1} is the convex hull of a subset of the vertices of its parent P_i. P_{i+1} is formed by removing in turn each vertex in $V(P_{i+1}) - V(P_i)$. The cone of faces attached to the vertex is also removed. This leaves a hole in the polyhedron. This hole is retriangulated in convex fashion. By removing an independent set of vertices, we assure that the apexes of removed cones are not connected. As a result we can reattach one or many cones and retain a convex object. We only remove vertices of degree less than 12. As a result, the hierarchy has logarithmic depth. Computing the hierarchy requires linear time and it can be stored in linear space. The initial segment of the video shows the creation of a 6 level hierarchy from a convex polyhedron built on 30 vertices. Colors are used to code the levels .

The hierarchy has been constructed as a search tool. We demonstrate its use in searching during the final segments of the video. We begin with the task of determining if a polyhedron intersects a plane. This is done by determining if the polyhedron at each level of the hierarchy intersects the plane and finding the closest vertex if there is no intersection. At the initial level, this computation can be done in constant time by enumeration. The hierarchy was constructed to simplify growth between levels. In particular, we need only consider growth affecting two edges adjacent to the closest vertex. This amounts to a maximum of 4 cones reattached at each level. Since the cones are formed from vertices of

*This work supported in part by the National Science Foundation under Grant Number CCR90-02352 and by The Geometry Center, University of Minnesota, an STC funded by NSF, DOE, and Minnesota Technology, Inc.

bounded degree, this growth can be done in constant time, also. As a result, a preprocessed polyhedron's intersection with a plane can be detected in $O(log\ n)$ time. We trace the hierarchical growth by highlighting the closest vertex and extremal edges. We also display the separating plane. Watching the cones return to the polyhedron we are able to see the algorithm determine if a near vertex is closest or new edges are extremal.

This video was prepared in the computer science department at Princeton University. Animation software was attached to code for computing the hierarchy and detecting intersections. The programs were written in C to run under UNIX on a Silicon Graphics Iris. Recording was done at the Interactive Computer Graphics Lab at Princeton and editing was done with the assistance of the Princeton Department of Media Services.

We thank Copper Giloth for helping with various artistic aspects of the animation. Kirk Alexander and Mike Mills led us through many difficult tasks in helping us produce the final video.

References

[1] D. Dobkin and D. Kirkpatrick, Fast detection of polyhedral intersections, *Journal of Algorithms*, 6, 381-392, 1985.

[2] D. Dobkin and D. Kirkpatrick, Determining the separation of preprocessed polyhedra – a unified approach, ICALP-90, LNCS 443, 400-413, Springer Verlag, Berlin, 1990.