To appear: Technical Sketches, SIGGRAPH 2001

Efficient Update of Geometric Constraints in an Evolving Mesh Representation

Maryann Simmons Sara McMains Carlo Séquin
U.C. Berkeley

1 Introduction

This sketch describes the use of motion bounds to optimize the evolving mesh representation used by the *tapestry*[1] interactive rendering system. The tapestry dynamic display mesh is constrained to be a "spherical depth mesh" (i.e. its projection onto a sphere centered at the viewpoint has no overlapping elements). The mesh vertices are projected onto a sphere centered at the viewpoint to determine the mesh topology during incremental insertion of new points. This makes the insertion more efficient by reducing the problem to 2D. A Delaunay condition is maintained on the projected mesh to produce a good quality image reconstruction, as well as guarantee robustness of the meshing code. In addition to the depth mesh and Delaunay constraints, minimum and maximum projected edge length, and minimum projected vertex-edge separation constraints are maintained on the mesh to provide an efficient and robust implementation. This sketch describes an efficient method for enforcing these constraints during view motion.

2 Calculating Conservative Motion Bounds

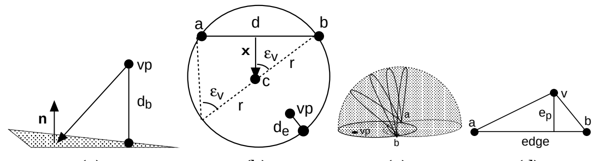
The mesh is constructed relative to an initial viewpoint. The geometric constraints are enforced by construction for this viewpoint. When the viewpoint changes, we want to retain as much of the 3D mesh as possible, while guaranteeing that all of the constraints are still met. Since it is inefficient to test each triangle to make sure that its projected topology remains consistent, we assign conservative motion bounds to each triangle stating how far the user can move before any constraints are invalidated.

For the depth mesh constraint, this means that all triangles that will become back-facing in the new view need to be removed (by deleting adjacent samples). We use the distance from the viewer to the plane of the triangle as a conservative bound d_b indicating how far the viewpoint could move before the triangle could be back-facing (see figure a).

For minimum projected edge length we note that in the plane, all points for which the angular extent of an edge (a, b) relative to a viewpoint is greater than some minimum ϵ_v are contained in the circle that has (a, b) as a chord, and an interior angle of ϵ_v (figure b). The same construction holds in 3D for the ϵ_v surface, which can be visualized by sweeping the circle about the edge in the half-space above the plane of the triangle to form a half torus with negative interior radius (figure c). To determine the motion bounds for edge length, we calculate the minimum distance d_e to the perimeter of this surface.

For minimum edge-vertex separation, the same calculations described in the previous paragraph are performed, but the perpendicular e_p dropped from the vertex to edge (a, b) is used as input instead of edge (a, b) (figure d).

Maximum edge length is maintained by inserting base vertices corresponding to an icosahedral subdivision of the sphere at the new view.



(a) (b) (c) (d)

3 Results

We have utilized this update technique to evolve a fully dynamic display mesh used as a front end for ray-tracing[2]. We further optimize the identification of non-conforming triangles by binning the mesh triangles based on the conservative motion bounds. In this way only a small number of triangles need to be examined each frame, and an even smaller number tested. During an interactive session with a moving observer, and a mesh with approximately 22K triangles, on average only 9 percent of the triangles had to be examined, and of those only 4 percent needed to be removed. The end result is that on average over 99 percent of the mesh could be re-used across view motions, while still maintaining the depth mesh and minimum feature size constraints.

To ensure a robust mesh and good quality image reconstruction, the Delaunay condition relative to the new view must also be re-asserted for each triangle. This requires that each sample be re-projected relative to the new viewpoint. As these operations are expensive, in this application we have relaxed the Delaunay constraint for efficiency. Instead, samples are re-projected, and the Delaunay condition tested, only when triangles are encountered in some subsequent mesh operation. The derivation of a similar motion bound for triangles based on the Delaunay condition would greatly improve the performance and mesh quality after motion.

References

[1] Simmons, M. and C. H. Séquin. "Tapestry: A Dynamic Mesh-based Display Representation for Interactive Rendering." Proc. of Eurographics Rendering Workshop, 1999.

[2] Simmons, M.. "Tapestry: An Efficient Mesh-based Display Representation for Interactive Rendering." PhD thesis. U.C. Berkeley May 2001.