HLODs: Hierarchical Levels of Detail

Hierarchical Simplification for Faster Display of
Massive Geometric Environments

Department of Computer Science University of North Carolina at Chapel Hill February 2004

The Challenge

We desire a way to interactively view very large polygonal models, such as those generated in CAD applications (Figure 1). As part of the Walkthrough group, one approach we have investigated extensively is the use of geometric approximations or levels of detail (LODs). LODs are error-bounded simplified versions of a model that can be displayed more quickly.

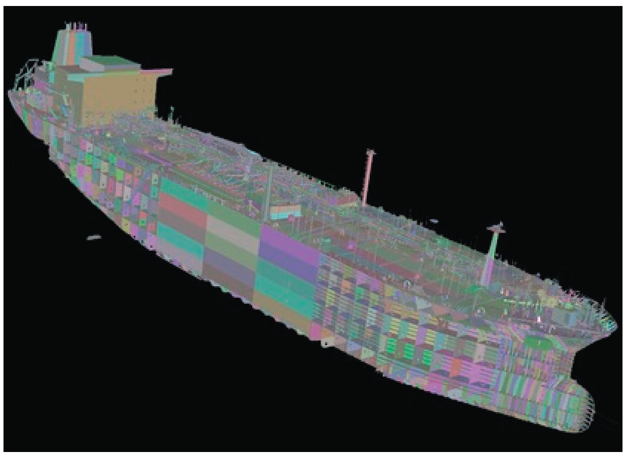

Figure 1. The Double Eagle tanker. The model has 126,630 objects and 82,361,612 triangles.

Highlights

- **Fidelity: We group objects to create HLODs, thus increasing the visual quality of drastic approximations.**
- **Automatic Generation: We compute the HLODs of a scene graph without user intervention.**
- **Generality: We make no assumptions about topological information or representation.**
- **Efficiency: HLODs are static, so they can be rendered using display lists.**
- **Flexibility: Our HLOD scene graph structure allows us to render in constant frame-rate mode or image-fidelity mode.**

Overview

Traditional LOD methods can typically only simplify a single object at a time. Given a scene containing multiple objects, these methods can only minimize errors local to individual objects and not the scene as a whole.

Hierarchical LODs, or HLODs, are generalizations of traditional LOD methods to hierarchical aggregations of objects. We generate HLODs by simplifying separate portions of a scene together, producing higher fidelity global approximations, especially for drastic simplifications. Thus for a target number of polygons, LODs might not render all objects in a scene, producing cracks (Figure 2). HLODs give a more solid approximation (Figure 3).

For dynamic scenes, we dynamically recompute HLODs asynchronously, taking advantage of parallel hardware when available.

Creating HLODs

While traditional LODs represent the geometry of a single node in the environment, HLODs represent entire branches of the scene graph, or the geometry of multiple nodes.

We represent the environment using a scene graph and first compute standard LODs for each node. We then compute HLODs at each node in a bottom-up fashion based on those LODs:

- The HLODs of a leaf node are equivalent to its LODs.
- The HLODs of an internal node in the scene graph are computed by combining any LODs belonging to the node itself with the HLODs of its children.

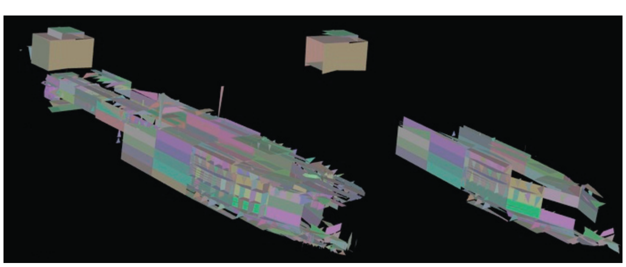

Figure 2. LODs of the Double Eagle. They consist of 7,887 and 1,922 faces.

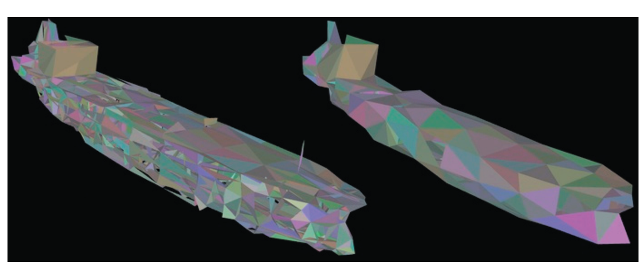

Figure 3. HLODs of the Double Eagle. They consist of 7,710 and 1,914 faces.

Rendering HLODs

In a traditional LOD rendering system, the display algorithm will render an appropriate level of detail for every object or node in the scene graph. Since an HLOD of a node is an approximation of the geometry of that node as well as its descendants, if we render a node's HLOD while traversing the scene graph, we do not need to visit its descendants (see Figure 4). In this way we can aggressively cull out entire portions of the scene by merely rendering the HLODs at a node when appropriate. In Image Quality Mode, we render by traversing the scene graph until the error bound associated with an HLOD satisfies the projected screen-space error constraint. In Target Frame Rate Mode, we refine the coarsest HLODs with maximum projected error until any further refinements would cause us to render more polygons than time permits.

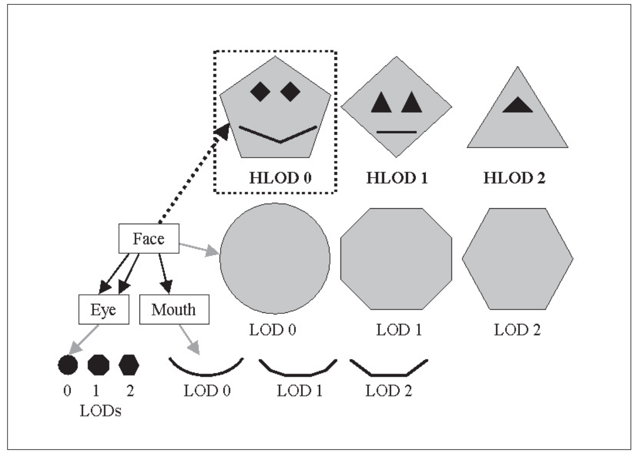

Figure 4. Rendering a face model using LODs and HLODs. Our algorithm traverses the scene graph starting at Face. Since the viewer is far away, it renders a representation of Face using HLOD 0. Since this HLOD represents the entire scene graph, the system stops the traversal.

Ph.D. Researcher

Carl Erikson

Thesis Adviser

Dinesh Manocha, professor

Graduate Research Assistants

William Baxter, Wesley Hunt, Avneesh Sud

Research Sponsors

Honda
Intel Corp.
National Institutes of Health, National Center for Research Resources
National Science Foundation
NSF/ARPA Center for Computer Graphics and Scientific Visualization
Office of Naval Resarch
U.S. Army Research Office

Fellowships: Intel Fellowship, NCAA Graduate Fellowship, NSF Graduate Fellowship, Alfred P. Sloan Foundation Fellowship.

Selected Publications

Erikson, C., D. Manocha, and W. Baxter. "HLODs for Faster Display of Large Static and Dynamic Environments," *Proc. 2001 Symposium on Interactive 3D Graphics*, March 2001.

Erikson, C. Hierarchical Levels of Detail to Accelerate the Rendering of Large Static and Dynamic Polygonal Environments, Doctoral Dissertation, Department of Computer Science, University of North Carolina at Chapel Hill, 2000.

Erikson, C., and D. Manocha. "GAPS: General and Automatic Polygonal Simplification," *Proc. 1999 Symposium on Interactive 3D Graphics*, 1999, 79–88, 225.

Erikson, C., and Manocha, D. "Simplification Culling of Static and Dynamic Scene Graphs," Department of Computer Science technical report TR98-009, University of North Carolina, 1998.

Acknowledgements

- The Power Plant environment is courtesy of James Close and Combustion Engineering, Inc.
- The Ford Bronco dataset is courtesy of Viewpoint and Division, Inc.
- The Double Eagle model is courtesy of Rob Lisle, Bryan Marz, and Jack Kanakaris at Newport News Shipbuilding.

For More Information

Dr. Dinesh Manocha
Department of Computer Science
University of North Carolina at Chapel Hill
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175
Phone: (919) 962-1749
Fax: (919) 962-1799
E-mail: dm@cs.unc.edu
Web: www.cs.unc.edu/~walk/hlod