Quick-VDR: Interactive View-Dependent Rendering of Massive Models

Sung-Eui Yoon Brian Salomon Russell Gayle Dinesh Manocha
University of North Carolina at Chapel Hill
{sungeui,salomon,rgayle,dm}@cs.unc.edu
http://gamma.cs.unc.edu/QVDR

Introduction Recent advances in acquisition, modeling, and simulation technologies have resulted in large databases of complex geometric models. These gigabyte-sized datasets consist of tens or hundreds of millions of polygons and pose a number of challenges for display and manipulation.

View-dependent simplification and rendering have been actively researched for interactive display of large datasets [Hoppe 1997]. These algorithms have many appealing properties because they compute different levels-of-detail (LODs) over different regions of the model using from a precomputed hierarchy of simplification operations. The selection of LODs occurs at runtime based on the viewing and lighting parameters. This reduces the "popping" artifacts that can occur while switching between different LODs in systems based on static LODs [Erikson et al. 2001]. The algorithms generally maintain a cut, or *active vertex front*, across the hierarchy which is traversed each frame for mesh refinement.

Current representations and refinement algorithms for view-dependent rendering do not scale well to large models composed of tens or hundreds of millions of triangles. The refinement cost is a function of the front size and may be prohibitively expensive for massive models. Furthermore, resolving dependencies in the vertex hierarchy can be expensive.

In addition to reducing the refinement cost, it is necessary to integrate view-dependent simplification algorithms with occlusion culling and out-of-core rendering. Algorithms for occlusion culling and out-of-core techniques also perform computations based on the view parameters. However, no known algorithms integrate conservative occlusion culling and out-of-core rendering with vertex hierarchies.

Our Approach We have developed a new view-dependent rendering algorithm (Quick-VDR) for interactive display of massive models. We use a novel scene representation, a *clustered hierarchy of progressive meshes* (CHPM). The cluster hierarchy is used for coarse-grained view-dependent refinement. The PMs, represented as a linear list, provide fine-grained local refinement to reduce the popping between successive frames without high refinement cost because PM refinement is relatively fast.

At runtime we maintain a cut of clusters in the hierarchy each of which contains a PM that is refined based on the view parameters. Our rendering algorithm uses temporal coherence and occlusion queries for visibility computations at the cluster level. The set of previously visible clusters is used as a potentially visible set (PVS) for occlusion culling in the current frame. For out-of-core rendering we prefetch for LOD changes and add a frame of latency in the overall pipeline for fetching newly visible clusters.

Quick-VDR relies on an out-of-core algorithm to compute a CHPM that performs a hierarchical cluster decomposition and simplification. Previous approaches such as [Hoppe 1998; Erikson et al. 2001] simply constrain simplifications of shared vertices along the boundary, but with a very deep hierarchy used for massive models, the constraints can result in a high triangle count simplification. We introduce the concept of *cluster dependencies* between adjacent clusters to generate drastic crack-free simplifications of the original model. The intuition behind cluster dependencies is that precomputed simplification constraints on shared vertices can be replaced by runtime cluster dependencies.

We have implemented and tested Quick-VDR on a commodity PC with NVIDIA 5950FX Ultra card. To illustrate the generality of our approach we have highlighted its performance on several models: a complex CAD environment (12M triangles), scanned models (372M triangles), and an isosurface (100M triangles) (Fig. 1). We can render these models at $10-35$ frames per second using a limited memory footprint of $400-600$MB.

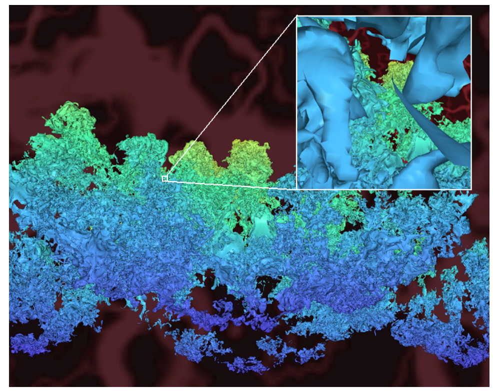

Figure 1: Isosurface. *These images show the application of Quick-VDR to a complex isosurface (100M triangles) generated from a very high resolution 3D turbulence simulation. The isosurface has high depth complexity, holes, and a very high genus. Quick-VDR can render it at 11 - 21 frames per second on a PC with NVIDIA GeForce FX5950 card and uses a memory footprint of 600MB. Model courtesy Mark Duchaineau and Peter Lindstrom, Lawrence Livermore National Laboratory.*

Advantages of Quick-VDR Quick-VDR integrates view-dependent simplification, conservative occlusion culling, and out-of-core rendering for high quality interactive display of massive models on current graphics systems. As compared to prior approaches, Quick-VDR offers the following benefits:

1. **Lower refinement cost:** The overhead of view-dependent refinement in the CHPM is one to two orders of magnitude lower than vertex hierarchies for large models.
2. **Massive Models:** We compute drastic simplifications of massive models, using hierarchical simplification with cluster dependencies, necessary for interactive rendering.
3. **Runtime performance:** Quick-VDR renders CHPMs using a bounded memory footprint and exploit the features of current graphics processors to obtain a high frame rate.
4. **Image quality:** We significantly improve the frame rate with little loss in image quality and alleviate popping artifacts between successive frames.
5. **Generality:** Quick-VDR is a general algorithm and applicable to all types of polygonal models, including CAD, scanned, and isosurfaces.

Acknowledgements Supported in part by ARO Contract DAAD19-02-1-0390, NSF awards ACI 9876914, NSF ACR-0118743, ONR Contracts N00014-01-1-0067, and Intel.

References

ERIKSON, C., MANOCHA, D., AND BAXTER, B. 2001. Hlods for fast display of large static and dynmaic environments. *Proc. of ACM Symposium on Interactive 3D Graphics*.

HOPPE, H. 1997. View dependent refinement of progressive meshes. In *ACM SIGGRAPH Conference Proceedings*, 189–198.

HOPPE, H. 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *IEEE Visualization Conference Proceedings*, 35–42.