Virtually Infinite Resolution Deformable Surfaces

Michael B. Nielsen[1], Ola Nilsson[2], Andreas Söderström[2], and Ken Museth[1,2]

[1]University of Århus [2]Linköping University

1 Introduction and previous work

We present a novel framework that allows for level set simulations at virtually infinite resolutions - the only limitation being the available disk space. The key components are transparent layers that combine fast out-of-core data management with efficient compression. For out-of-core simulations our framework is significantly faster than OS-based virtual memory schemes. In addition, extensive tests have shown our framework to sustain throughputs that are about 65% of current state-of-the-art for in-core simulations.
A level set is essentially a time-dependent implicit geometry representation and has proven very popular for deforming surface problems. Several data structures for high resolution level sets and fluids have recently been proposed (*e.g.* DT-Grid[Nielsen and Museth 2006] and H-RLE[Houston et al. 2006]). However, all of these approaches are effectively restricted by the main memory available - typically a few GB. Common to these state-of-the-art grid representations is that they *separately encode grid topology and values*. Our framework can transparently be integrated with any of these grids, thus requiring minimal transition overhead in an existing pipeline. In addition it utilizes the virtually infinite size of today's disks or arrays, as well as reduces disk space and memory footprints by means of efficient compression. Many important graphics applications can directly benefit from our work. We present several examples, including a level set of resolution $\approx 30000^3$ which we believe to be an order of magnitude larger than ever demonstrated before.

2 Out-of-core and compression framework

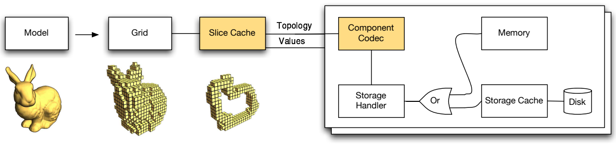

Figure 1: Our framework for out-of-core and compressed level sets and fluids.

An overview of our framework is illustrated in figure 1. From left to right: The level set or fluid is represented on a sparse 3D computational *Grid*. The *Slice Cache* temporarily buffers 2D slices of this grid needed to compute finite differences during simulations. The rest of the grid is stored compressed in-core or out-of-core. The Slice Cache provides both efficient sequential and random access to grid points in a local stencil. As the simulation computations progress through the grid, the slices are replaced and streamed, possibly compressed, to and from memory and disk. The staggered boxes shown on the right illustrate that the *Topology* and numerical *Values* of the grid are processed separately. This can for instance be utilized for ray-tracing where the topology can be kept in-core and numerical values stored out-of-core. The *Component Codecs* take advantage of application specific knowledge to obtain good compression. The topology is compressed using statistical encoding with local predictions based on Lagrange polynomials and planar approximations. Context-based encoding further increases the compression. The numerical values of the grid are compressed using multi-dimensional differential encodings which we show to out-perform existing high-preformance codecs. A *Storage Handler* stores the grid components in memory or on disk, and a *Storage Cache* efficiently implements our out-of-core scheme by combining novel prefetching and page-replacement techniques optimized for the access patterns of level set and fluid simulations. The components of our framework can be combined arbitrarily which allows for maximal flexibility for a particular application.

Figure 2: Fluid animation using our out-of-core framework.

3 Applications

Fluid Animations: The out-of-core framework allows us to significantly increase the resolutions of fluid animations beyond the memory available. In our current implementation particles, surface velocities and DT-Grid based level set surfaces and boundaries are stored out-of-core. Fluid velocities and pressure are kept uncompressed in-core. Fig. 2 shows a fountain simulation using a statue with an effective resolution of 512^3 as the boundary. The peak memory footprint for this out-of-core simulation was 711 MB compared to an estimated 3.7GB if run in-core. Using a standard ATA harddrive the throughput of the out-of-core simulation was about half of the throughput of an in-core simulation.

Surface Deformations: We have created a detailed level set CSG bunny (1.62GB) modeled as an out-of-core CSG intersection of a 2048^3 bunny (304MB) with a tiling of 20^3 smaller bunnies each at resolution 128^3 (7.47GB). Sizes are in uncompressed DT-Grid format. An out-of-core morph between the bunny and the CSG bunny that required close to 5GB of storage was run on a PC with 1GB memory and is shown in the supplementary video. The throughput was equal to 65% of the DT-Grid throughput for in-core simulations. Using an efficient out-of-core scan converter we have generated level sets of resolution $\approx 30000^3$ and requiring 24GB storage.

PDEs on Surfaces: We have run several simulations directly on the level set manifold. In the video we show the wave equation on models of sizes above 1024^3 requiring several GB of memory.

References

HOUSTON, B., NIELSEN, M., BATTY, C., NILSSON, O., AND MUSETH, K. 2006. Hierarchical RLE Level Set: A Compact and Versatile Deformable Surface Representation. *ACM Transactions on Graphics 25*, 1, 1–24.

NIELSEN, M. B., AND MUSETH, K. 2006. Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets. *Journal of Scientific Computing ISSN: 0885-7474*.