DB+Grid: A Novel Dynamic Blocked Grid For Sparse High-Resolution Volumes and Level Sets

Ken Museth, *DreamWorks Animation, SKG*

Figure 1: Enright benchmark test on high-resolution DB+Grid. Divergence-free advection of the Armadillo at an effective resolution of 4096^3. Far-left: Close-up of dynamic 4^3 blocks. Frames show level sets at time= $0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1$ period. Notice how the initial (time=0) and last (time=1) frames appear almost identical, indicating low mass-loss due to numerical dissipation. This is a consequence of the high resolution.

Introduction

We have developed a new data structure for the efficient representation of sparse, time-varying volumetric data discretized on a 3D grid. Our "DB+Grid" , so named because it is a Dynamic Blocked Grid that shares several characteristics with B+Trees (typically employed in databases and file systems), exploits the spatial coherency of time-varying data to separately and compactly encode data values and grid topology. DB+Grid allows for cache-coherent and fast data access into sparse, 3D grids of very high resolution, exceeding millions of grid points in each dimension. Additionally, DB+Grid is very general, since it imposes neither topology restrictions on the sparsity of the volumetric data nor access patterns when the data are inserted, retrieved or deleted. This is in contrast to most existing sparse volumetric data structures, which either assume static data (*i.e.* values and topology) or fixed data topology (*e.g.* manifold surfaces) and require specific access patterns to avoid slow random access. Since DB+Grid is a hierarchical data structure, it also facilitates adaptive grid sampling, and the inherent acceleration structure leads to fast algorithms that are well-suited for simulations. As such, DB+Grid has proven useful for several applications that call for very large, sparse, animated volumes, *e.g.* level sets and fluid simulations. In this talk we will compare DB+Grid with existing state-of-the-art dynamic sparse data structures and showcase applications from the visual effects industry.

Highlights

DB+Grid is a sparse and dynamic volumetric data-structure with the following characteristic features:

Dynamic. Unlike most sparse volumetric data structures DB+Grid is developed for both dynamic topology and values typical for time-dependent numerical simulations and animated volumes. This requires efficient implementation of *sparse* finite-difference iterators, topology morphology (*e.g.* dilation) and rebuild algorithms as well as temporal value buffers for cache-coherent numerical integration.

Compact. A dynamic and hierarchical allocation of blocks leads to a compact sparse data-structure that allows for extremely high grid resolutions. To further reduce the memory-footprint we present an efficient, lossless compression technique that can be employed for both on-line and off-line storage. To further reduce the off-line footprint we support a combination of bit-quantization and standard compression schemes.

General topology. Unlike most existing dynamic data structures for narrow-band level sets ours can effectively represent sparse volume data with arbitrary dynamic topology. This implies that DB+Grid can be use as a generic volumetric data structure as oppose to merely supporting dynamic level set applications.

Fast and flexible data access. DB+Grid supports fast constant-time random data lookup, insertion and deletion. DB+Grid also offers fast (constant-time) sequential stencil access iterators which are essential for efficient simulations employing finite-difference schemes. Spatially coherent access-patterns even have an amortized computational complexity that is independent of the depth of the underlying B+tree employed by the DB+Grid.

Efficient Algorithms. Our hierarchical blocking approach offers several benefits, including; cache-coherence, inherent bounding-volume acceleration and fast per-block (vs per-voxel) operations. It also lends itself to several hierarchical optimization techniques resulting in improved computational performance, *e.g.* near real-time boolean operations and efficient multi-threading schemes.

Adaptive Grid. Unlike most existing narrow-band level set data structures DB+Grid is hierarchical and can store values at adaptive resolution, by encoding data at all levels of the underlying B+tree. While obviously no as adaptive as octrees, this feature is very useful for applications like ray-marching and collision detection of sparse volumes, *e.g.* narrow-band level sets.

Configurable. By design DB+Grid is highly configurable in terms of tree-depth, fan-out factors and block dimensions, even wrt. the 3 coordinate axis. This allows the grid to be tailored to specific applications in order to optimize factors like memory foot-prints, cache utilization, random vs. sequential access times and grid adaptively.

Out-Of-The-Box. The domain of DB+Grid is virtually unbounded in the sense that it can dynamically and randomly expand and contract in all eight coordinate directions without requiring a full and expensive re-allocation and deep copy. This is an especially desirable property for grids with dynamic topology. The support for negative grid indexing has also proved convenient for many practical applications.

Out-of-core. DB+Grid supports simple out-of-core streaming. More specifically, we can reduce the in-core memory footprint by storing grid values out-of-core and only keep the grid topology in-core. Values are then loaded on demand *e.g.* during ray-tracing.