Novel Algorithm for Sparse and Parallel Fast Sweeping: Efficient Computation of Sparse Signed Distance Fields

Ken Museth
DreamWorks Animation

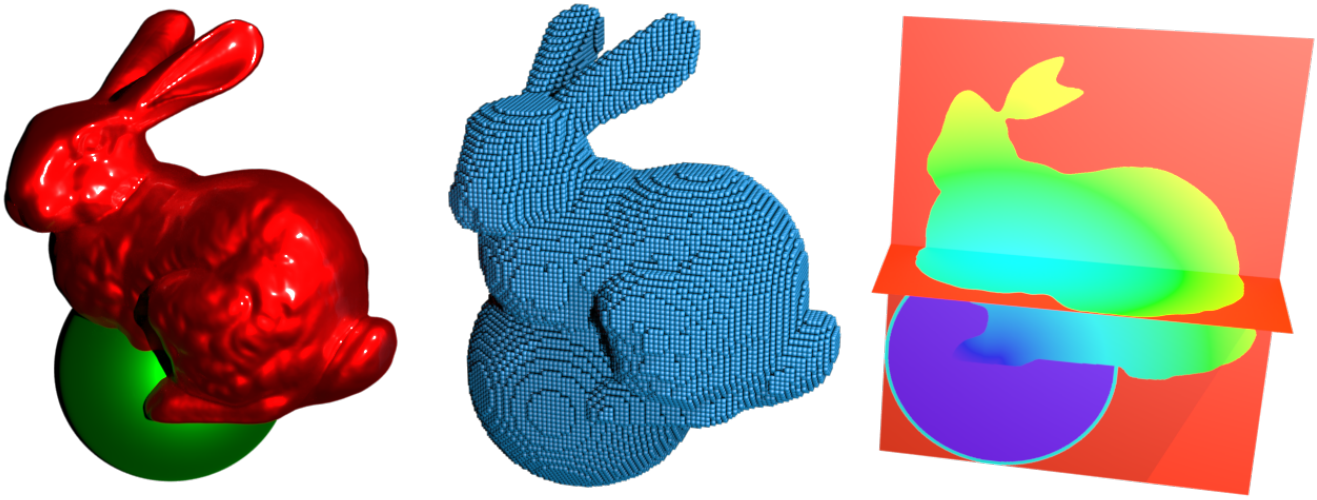

Figure 1: Extending a sparse signed distance field from the narrow band of a level set sphere, into an overlapping mask defined as a bunny. Left: Surfaces of the overlapping (green) sphere and the (red) bunny, act as respectively the boundary condition and mask for our Sparse and Parallel Fast Sweeping Method. Middle: Illustration of the leaf nodes of the underlying sparse VDB trees. Right: Orthogonal cross-sections of the extended sparse signed distance field. Note that while these cross-sections are dense, the volumes are sparse. The bunny has an efficient voxel resolution of $628x621x489$ and the extended level set function took about two seconds to compute on a workstation with 24 CPU cores.

ABSTRACT

We present a new efficient algorithm for computing signed distance fields by means of the Fast Sweeping Method. Unlike existing algorithms ours is explicitly designed to explore the benefits of sparse (vs dense) grids as well as concurrency, i.e. mutli-threading.

CCS CONCEPTS

•Computing methodologies →Concurrent algorithms;

KEYWORDS

level sets, signed distance fields, fast sweeping, eikonal equation.

ACM Reference format:
Ken Museth. 2017. Novel Algorithm for Sparse and Parallel Fast Sweeping: Efficient Computation of Sparse Signed Distance Fields. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085093

MOTIVATION

Signed distance fields (SDFs) are fundamental to numerous applications across many different disciplines, including of course computer graphics. To mention just a few such examples consider most level set applications, velocity extension, and collision detection. Our driving application was an efficient way for artists to perform attribute transfer from one surface onto a background grid or even another surface. As it turns out, this hinges on a fast algorithm to compute a sparse SDF in a user-defined sub-space given fixed boundary conditions.

Mathematically speaking, SDFs are Euclidean distance functions, $\phi(\vec{x})$, which are solutions to the Eikonal equation, $|\nabla\phi(\vec{x})| = F(\vec{x})$, with a unit speed function, $F(\vec{x}) = 1$, and specific boundary conditions, $\{\vec{x}|\phi(\vec{x}) = 0\}$. Given the ubiquity of SDFs it is only natural that several different algorithms have been proposed to solve the Eikonal equation. Of these the most celebrated is arguably the Fast Marching Method (FMM) [Sethian 1996] which is closely related to Dijkstra's algorithm. A less well known, but computationally superior algorithm, is the so-called Fast Sweeping Method (FSM) [Zhao 2004] which offers $O(N)$ vs $O(N \log N)$ time complexity where N is the number of grid points for which the Eikonal equation is solved.

Recent years have seen the advance of a sparse grid dubbed VDB[Museth 2013], which is a compact data structure and toolset for high-resolution volumetric effects typically encountered in movie production. Since its open source release in 2012, as OpenVDB[1], it has been adopted by many of the major third-party renders and tools in the VFX industry. Thus, it should come as no surprise that it is highly desirable to have efficient algorithms for computing and extending SDFs on non-dense grids like VDB. While FMM is easily adopted to sparse grids, it is challenging to parallelize since

SIGGRAPH '17 Talks, Los Angeles, CA, USA
 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085093

[1] http://www.openvdb.org

it typically employs a dynamically updated min-heap tree data structure. Conversely FSM is relatively easy to multi-thread, e.g. by means of domain-decomposition, but to the best of our knowledge it has never been re-formulated to take full advantaged of sparse grids. This dichotomy is precisely what motivated our work.

DENSE FAST SWEEPING METHODS

Fast Sweeping Methods take point of reference in some well known facts about the the Eikonal equation, namely that it is a non-linear hyperbolic partial differential equation that can be solved numerically a by means of the following single-sided (so-called up-wind) finite difference scheme due to Godunov

$$max(\phi(i,j,k) - min(\phi(i-1,j,k), \phi(i+1,j,k)), 0)^2 + \quad (1a)$$

$$max(\phi(i,j,k) - min(\phi(i,j-1,k), \phi(i,j+1,k)), 0)^2 + \quad (1b)$$

$$max(\phi(i,j,k) - min(\phi(i,j,k-1), \phi(i,j,k+1)), 0)^2 = \Delta h^2 \quad (1c)$$

where $\phi(i,j,k)$ denotes the discretization of $\phi(\vec{x})$ at the grid point $\vec{x} = (i\Delta h, j\Delta h, k\Delta h)$, and Δh is the uniform spacing of grid points. Eq. (1) is a quadratic equation in $\phi(i,j,k)$ that can be solved locally using standard techniques. It should be evident that it couples the values at the grid point (i,j,k), $\phi(i,j,k)$, with its six nearest neighbor points along the axial directions (possibly not all at the name time). As in the original implementation of FSM, [Zhao 2004], the grid values, $\phi(i,j,k)$ are initialized to the boundary values $\{\vec{x}|\phi(\vec{x}) = 0\}$ where available, and is set to $\pm\infty$ everywhere else, where the sign is derived from the boundary values.

Due to the hyperbolic nature of the Eikonal equation, information can arrive at a given grid point from any direction, and it is exactly the job of the Godunov scheme, Eq. (1), to "automatically select" the correct direction by selecting the closest neighboring grid points, i.e. the ones with the minimum value, since this exactly corresponds to the Euclidean distance to the known values defined by the boundary conditions. This observation is the crux of the FSM [Zhao 2004] which leads to an iterative Gauss-Seidel method where Eq. (1) is solved repeatedly by alternating sweeps (hence the name) over the grid points, (i,j,k) that covers all the possible groupings of directions from which information can flow on the computational grid. In 3D this corresponds to $2^3 = 8$ directions (or sweeps), which is sufficient for a global solution to the Eikonal equation if the characteristic directions are not crossing, i.e. no shocks. However, within each such a sweep the grid points are solved for in a fixed sequential order, ultimately resulting in a sequential algorithm with a computational complexity that is linear in the number of grid points.

In the case of dense grids it is fairly easy to achieve a parallel FSM algorithm, for instance by means of domain-decompositions and halo-exchanges as outlined in [Zhao 2007]. More recently [Detrixhe et al. 2013] made use of an interesting observation first noted in the original publication of [Zhao 2004] (c.f. figure 4.1(a)), namely that the domain of dependency of grid points within a given sweep direction is grouped in dense planes angled relative to the three axial directions. This can be seen as a consequence of the fact that the up-wind finite difference stencil employed in Eq. (1) only involves grid points in the three axial directions, and thus solutions are independent between neighboring grid points that are off-axis. [Detrixhe et al. 2013] utilized this to visit grid points arranged in slices through the a dense volume, and then process them concurrently as in [Zhao 2004]. Since they also assumed the computational grid to be dense, defining these planes was relatively easy, and in the end the final algorithm, while elegant, was surprisingly close the original implementation in [Zhao 2004].

OUR SPARSE FAST SWEEPING METHOD

The observations outlined above form the foundation for our work, but unfortunately the leap from dense grids to sparse grids is not trivial. For starters, how to efficiently determine the correct ordering of the sparse grid points that corresponds to the eight sweep directions, as well as the grouping of grid points with independent domains of dependency, which facilitates parallel computation?

Our solution, which was developed and optimized specifically for the OpenVDB sparse data structure makes use of a new thread-safe paged data structure to concurrently define the sparse grid points, as well as fast sorting algorithms that effectively achieve the grouping of grid points into the independent planes first noted in [Zhao 2004] and later utilized in [Detrixhe et al. 2013] for dense grids. Specifically, we sort the sparse grid points (i,j,k) based on four fast to compute sort-keys, $(i+j+k)$,$(i+j-k)$,$(i-j+k)$, and $(i-j-k)$, that define sweeping planes in which solutions to Eq. (1) are independent and therefore can be processed concurrently. This new sorting step incurs a relatively small overhead since it can be implemented as an unstable parallel sort, and a single sort can be reused for two sweep directions, i.e. only four sorts are required to perform all eight FSM sweeps in three spatial dimensions. We pay careful attention to all our data structures and algorithms to ensure that they are fully multi-threaded, and have observed good scaling for most practical examples. We note that while our algorithm operates directly on the sparse VDB data structure (with no need for deep copies or intermediate value buffers) it should also be applicable to other sparse grids. Finally, our algorithm allows for user-defined masks of sparse grid points that restrict the SDF computations to regions of interest, further localizing and improving its performance.

CONCLUSION

We have developed a new concurrent algorithm for solving the Eikonal equation on sparse grids, that is when both the boundary condition, $\{\vec{x}|\phi(\vec{x}) = 0\}$, and the solution (SDF) to $|\nabla\phi(\vec{x})| = F(\vec{x})$ are represented on grids with sparse (vs dense) layouts of its computational nodes. Given the ubiquity of sparse grids, like OpenVDB, we believe this work to be of interest to other developers and practitioners alike, and we plan to make a full implementation available.

REFERENCES

Miles Detrixhe, FréDéRic Gibou, and Chohong Min. 2013. A Parallel Fast Sweeping Method for the Eikonal Equation. *J. Comput. Phys.* 237 (March 2013), 46–55. DOI: http://dx.doi.org/10.1016/j.jcp.2012.11.042

Ken Museth. 2013. VDB: High-Resolution Sparse Volumes with Dynamic Topology. *ACM Trans. Graph.* 32, 3, Article 27 (July 2013), 22 pages. DOI: http://dx.doi.org/10.1145/2487228.2487235

James A. Sethian. 1996. A fast marching level set method for monotonically advancing fronts. *Proc. of the National Academy of Sciences of the USA* 93, 4 (February 1996), 1591–1595.

Hongkai Zhao. 2004. Fast Sweeping Method for Eikonal Equations. *Math. Comp.* 74 (2004), 603–627.

Hongkai Zhao. 2007. Parallel implementation of the fast sweeping method. *Journal of Computational Mathematics* 25, 4 (2007), 421–429.