

The Split Grid - A Hierarchical 1D-Grid-based Acceleration Data Structure for Ray Tracing

Pablo Bauszat¹, Marc A. Kastner¹, Martin Eisemann¹ and Marcus Magnor¹

¹Computer Graphics Lab, TU Braunschweig, Germany

Abstract

We present a new acceleration structure for ray tracing called the Split Grid. Combining concepts of hierarchical grids, kd-trees and Bounding Volume Hierarchies (BVHs), our approach is based on the idea of nesting 1D-grids. Our proposed acceleration structure is compact in storage, adaptive to the scene geometry and can be traversed using a fast and efficient traversal scheme. We show that the Split Grid is comparable to other current state-of-the-art acceleration structures regarding traversal performance and memory footprint. While other data structures usually achieve these levels of performance only due to a complex and expensive construction process (e.g. using the Surface Area Heuristic (SAH) [MB90]), our proposed Split Grid is built with a very simplistic construction scheme which is a major benefit of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Ray tracing is a core technique for many current global illumination rendering algorithms and other types of physical simulation methods and many researchers have shown that acceleration structures are essential for fast ray tracing performance [Hav00]. Acceleration structures need to offer a fast traversal performance, low construction time and a small memory footprint in order to be considered efficient. With the rise of GPU-based and dynamic scene ray tracing, construction performance receives more focus and applications tend to stride away from long precomputation times.

The most well-known acceleration structures in ray tracing today are grids, kd-trees and Bounding Volume Hierarchies (BVH) (see [Hav00, WMG*09] for overviews). While kd-trees are considered to offer the fastest traversal performance, their building process is costly [WMG*09]. BVHs are inferior to kd-trees when it comes to traversal, but are today often used in dynamic ray tracing systems, mainly because their construction is considered to be faster and maps better to parallel architectures (e.g. GPUs). Recent research has investigated fast construction of kd-trees and BVHs on parallel architectures [WZL11, PL10] but they all require

a complex construction heuristic such as the Surface Area Heuristic (SAH) [MB90] to create high-quality hierarchies.

In the early beginning of ray tracing of dynamic scenes, uniform grids were investigated first due to their fast construction performance. While uniform grids can offer only linear traversal runtimes, recent research again focuses on hierarchical grid approaches for fast dynamic ray tracing on the GPU [KBS11]. However, by using only a two-level approach, the adaptiveness of the hierarchy is limited which can be a drawback in more complex scenes.

2. The Split Grid

We introduce the Split Grid as a new acceleration structure for ray tracing. It is a space-partitioning, adaptive and hierarchical acceleration structure and is based on the idea of hierarchical grids. Instead of recursively nesting complete 3D-grids at each recursion level, the Split Grid uniformly divides the scene-space only along a single axis into a variable number of grid cells of the same size and then recursively continues to subdivide these cells into 1D-Grids along a potentially different axis, Figure 1. Therefore, it resembles a kd-tree with multiple equidistant split planes. Additionally,

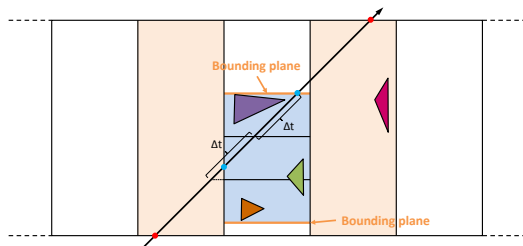


Figure 1: The Split Grid divides a single dimension of space into a number of uniformly sized grid cells in each subdivision step. The splitting axis as well as the resolution of resulting inner node can be optimized for every inner node. Additional bounding planes cut away empty space. The traversal resembles a mixture of kd-tree/BVH and grid traversal.

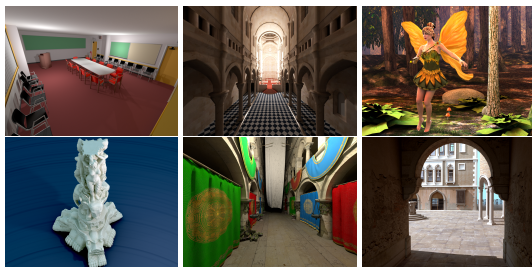


Figure 2: All scenes and statistics were generated for a resolution of 1024x768 pixels on an Intel Core i7-2600, 3.40 GHz and 16 GB RAM running Windows 7 64-bit. The complexity ranges from 76k to 10 million triangles.

each node stores two additional *bounding planes* orthogonal to the node's split axis to cut off empty space more efficiently and faster adaptation to the scene geometry. Whole sub-trees can be skipped immediately during traversal when rays move through empty space; a property usually reserved for object-space partitioning schemes. Overall, this allows fast traversal of the hierarchy using a mixture of kd-tree and grid traversal with sublinear runtime.

For the construction, we show that a low-priced construction algorithm can offer fast rendering performance without the use of any expensive cost functions, while the memory footprint is still comparable to other state-of-the-art acceleration structures. In each subdivision step the splitting axis is chosen to be equal to the longest axis of the cell. The resolution is based on a Gaussian function related to the primitive count to quickly subdivide the scene in the upper parts of the hierarchy and use a coarser subdivision towards the leaf nodes.

Figure 3 shows the traversal and construction times as well as the memory footprint of the Split Grid in comparison to the kd-tree, BVH and the classic grid for the scenes shown in Figure 2. The Split Grid achieves competitive traversal

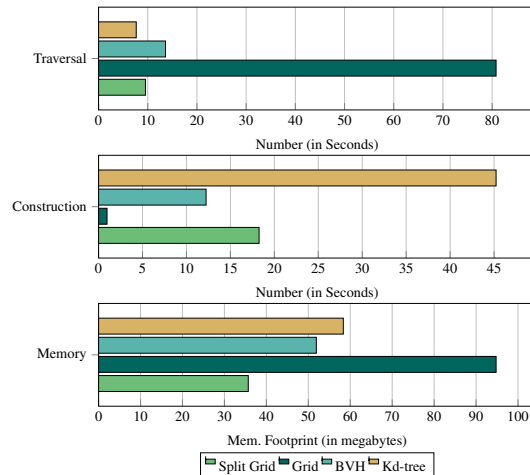


Figure 3: Comparison of traversal/construction time (in seconds) and memory footprint (in megabytes) of the Split Grid to current state-of-the-art acceleration structures. All statistics were measured for uni-directional path tracing without SIMD using 3 bounces of indirect illumination on the CPU and averaged over the 6 scenes of Figure 2.

performance while having the lowest memory footprint averaged over all scenes. The Split Grid outperforms the kd-tree during the construction phase, while achieving up to 66% of the performance of the BVH (where almost an order of magnitude less subdivision steps are performed due to object-space partitioning and the approximate, binned SAH heuristic). Note that all these results are achieved without the use of an expensive heuristic during the construction process. From a theoretical analysis and our premature results, we conclude that the Split Grid seems to be competitive to current state-of-the-art acceleration structures and in addition exhibits interesting properties indicating suitability for dynamic scenes and current GPU architectures.

References

- [Hav00] HAVRAN V.: *Heuristic Ray Shooting Algorithms*. Ph.d. thesis, Czech Technical University in Prague, 2000. 1
- [KBS11] KALOJANOV J., BILLETTER M., SLUSALLEK P.: Two-level grids for ray tracing on gpus. *Computer Graphics Forum* 30, 2 (2011), 307–314. 1
- [MB90] MACDONALD D. J., BOOTH K. S.: Heuristics for ray tracing using space subdivision. *Visual Computer* 6, 3 (1990), 153–166. 1
- [PL10] PANTALEONI J., LUEBKE D.: Hlbvh: hierarchical lbvh construction for real-time ray tracing of dynamic geometry. In *High Performance Graphics* (2010), pp. 87–95. 1
- [WMG*09] WALD I., MARK W. R., GÜNTHER J., BOULOS S., IZE T., HUNT W., PARKER S. G., SHIRLEY P.: State of the Art in Ray Tracing Animated Scenes. *Computer Graphics Forum* 28, 6 (2009), 1691–1722. 1
- [WZL11] WU Z., ZHAO F., LIU X.: SAH KD-tree construction on GPU. In *High Performance Graphics* (2011), pp. 71–78. 1