Introduction

While uniform grids are known for their fast build times, they historically suffer from slow ray tracing performance. With recent advances in the art, however, techniques such as micro 64-trees [1] and macro 64-regions [2] have helped to address this concern, especially with respect to rendering on GPUs. Though these works have also continued to improve the accelerator construction process, as of yet no attention has been given to their memory footprint. During construction, for instance, a significant amount of memory is consumed in auxiliary structures, and in the final output, two list pointers are used per voxel when only one is actually necessary. To tackle this problem, we improve on the build process proposed by Taranta and Pattanaik [2]; this is accomplished by employing additional atomic operations for synchronization between parallel tasks and by repeating certain calculations that were previously cached. See the Build Process (left).

Evaluation Results

To evaluate our build process, we tested twelve different scenes of varying complexity over a wide range of grid resolutions, ranging between 261,978 and 10,500,549 triangles. Further, all scenes were rendered at 1920 x 1280 using ray tracing of primary rays to ensure there is no loss in the performance of coherent rays. Further, each scene was rendered 1000 times with each iteration having a unique view (the camera position and direction are randomized in each iteration). The GPU was an NVIDIA GeForce GTX 680 GPU with 2048 MiB of memory. Results show a significant reduction in memory consumption (top right), from 20% to 45% over varying grid densities, and little change in the combined build and render performance (middle right).

References
