Assignment 2: BVH

Arian van Putten - 4133935

December 28, 2017

Originally, my plan was to improve the existing library in rust named `https://github.com/svenstaro/bvh`. However I ended up reimplementing the BVH from scratch, as there were lots of changes needed for the packed representation `https://github.com/arianvp/bvh/blob/master/src/packed_bvh/mod.rs`. However it is still a fork, and I might be able to turn it into a Proper PR with some work.

Surprisingly, at least on the provided micro benchmarks, the cache aligend traversal has not a lot of effect on the traversal performance. Once could argue whether it is even worth it. I did not test the original bvh in my ray tracer, so I can't tell if it had effect on the ray tracing performance.

However, ordered traversal and a stack-based approach did affect the performance a lot.j

1 Implemented features

- Stack based traversal (no recursion)
- Cache-line aligned allocator for Rust (It didn't have one)
- Packed BVH representation (nodes in 32 bytes)
- Ordered traversal
- Fast AABB-Ray intersections using SIMD
- Fast Binned construction with SAH along
- Ranged ray-packet traversal (using morton-curve ordering) has been implemented in the BVH (The function is named `interect_ranged` in `bvh/packed_bvh/mod.rs`) but is not used yet in the raytracer itself ,as I have not yet implemented sending packets of rays from the camera due to to time constraints :(
- Render Sponza scene with 2 frames per second.
- Render Mario 8 Wario Mountain on 4 to 10 frames per second.