Report project 2 - BVH, Advanced Graphics (2019)

Amber Elferink(5491525) & Ymke Wegereef (3971457)

Implemented functionality

We implemented a BVH using lighthouse 2 on the Whitted Raytracer implemented in assignment one with the following functionality:

- Binned building with the surface area heuristic (SAH)
  The method “Binning” computes and compares results for each of the three axis. It then chooses the best split plane based on the surface area heuristic. (There is also a method called “Partition that greedily compares the centroids of the triangles, based on their SAH, that returns as soon as an improvement is found.)
- Building a tree for the DeLorean (1082382 vertices in lighthouse = 360794 triangles) is completed within 0.556s (single core, 3.6 GHz). This is measured with the “Binning” method.
- Default BVH traversal, which renders the given DeLorean (360794 triangles, 1280x720px) in 180 ms quad core hyperthreaded (no threadpool) and in 2.45 seconds single core.

Division of work

For this assignment we’ve mostly worked together, as it was harder to split up the work on the BVH and as it was important for us both to fully understand the process. Hence we did a lot of pair programming.

Sources

- Ray-Box intersections: https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-box-intersection
- Our Binning method is based on the article: On fast Construction of SAH-based Bounding Volume Hierarchies, Wald, 2007

Report project 3 - Advanced Graphics (2020)

Amber Elferink(5491525) & Ymke Wegereef (3971457)

Implemented functionality

We converted our existing ray tracer into a path tracer (using lighthouse 2) with the following new functionality:

- Multiple importance sampling
- A photon map (with 1000000 photons) is build during initialization. Photons are stored in a hashed grid that is build around the scene.
- Photon based Next Event Estimation as described in the master thesis of Andreas Mikolajewski (see sources).
- Packet Traversal as described in Overbeck et al. in their paper “Large Ray Packets for Real-time Whitted Ray Tracing”. This was implemented for the whitted raytracer primary rays with frustra, for the pathtracer for the primary rays and the metallic rays. The intersection methods for both the nodes and the primitives are fully AVX.

One can switch between ray and pathtracer within the core_settings.h, although the light is a bit too strong in current scene for the raytracer.

Division of work

For this assignment we first implemented a basic path tracer together. Once this was finished, Ymke worked further on the importance sampling and PNEE and Amber worked on packet traversal.

Sources

- Efficient data structures and sampling of many light sources for Next Event Estimation, by Andreas Mikolajewski
- Large Ray Packets for Real-time Whitted Ray Tracing, by Overbeck et al.