CS 564 – PROJECT PROPOSAL

Reordering Linear Representation of Constrained Tetrahedralization in order to Further Accelerate Ray Tracing

Ray tracing is based on calculating many paths that are emitted from the direction of the camera into the scene. After many object interactions, paths that reach a light source contribute to the final image. For any ray that is cast into the scene, the algorithm needs to find the first polygon that this ray intersects with in each iteration. Since doing such line intersection tests for every polygon is not efficient, various methods are introduced in order to accelerate the operation.

One such method is “Accelerating Ray Tracing using Constrained Tetrahedralization”\(^{i}\), where the 3D space that includes the scene is divided into tetrahedral regions, in a way that each triangle face in the scene overlaps with a face of a tetrahedral. This makes it possible for a ray to check only the tetrahedral regions that it goes in for polygon intersection. For a ray that goes out of any tetrahedral region, it is possible to calculate the next tetrahedral region it goes in, which makes it possible to eliminate many unnecessary intersection tests.

However, linear representation of tetrahedralization of the 3D space is a large data structure, and only a subset of it could fit into the cache. In order to further accelerate the process, frequently used tetrahedral regions for a setup should have close proximity in the data set. This would require a special sorting of the tetrahedral regions, in a way that would minimize the distances of the used tetrahedrons in the linear representation. This project proposes the implementation of such algorithm, given a constrained tetrahedralization of a scene, that would generate a suitable arrangement where distances between regions of interest are minimized. Regions of interest are defined as the tetrahedral regions that a ray goes in or out in the 3D space, and a ray is expected to visit neighboring regions of any tetrahedral. Since a tetrahedron has 4 different neighbors, some of the neighbors should be favored in the linear representation, without distancing any two neighbors too much.

In order to test the algorithm, a path traced by a random ray could be used. Each successive pair of tetrahedrons that the ray passes would contribute with a distance to the sum. This distance is measured by the offset that successive tetrahedron has in the linear structure, to the other tetrahedron. The algorithm should decrease this sum relative to the given original arrangement, which would be acquired from TetGen\(^{ii}\).

\(^{i}\) https://pdfs.semanticscholar.org/1ce9/8e58efa8ad81eb019cf2944d87dd7e491c4b.pdf
\(^{ii}\) http://wias-berlin.de/software/tetgen/