Comparison of Voxel Ray Marching Acceleration Structures

Aaron Danton agrd2@cam.ac.uk

Supervisors

Day-to-Day Dounia Hammou
Marking Rafal Mantiuk

Director of Studies

Matthew Ireland

All core objectives have been met, which notably includes: implementation of the 5 acceleration structures (Grid, Texture, Octree, Contree, Brickmap); shader hot-reloading when a source file is edited; an ability to measure the performance metrics of those structures; scenes and structures can be changed during the runtime; modification for the grid, texture and brickmap. With the implementation of modification, key-frame animation support was also added.

To facilitate the use of larger models and key-frame animation, a separate program was written to convert meshes to voxels, and then handle the generation of the acceleration structures along with their serialization. This program also handles the per frame difference information to support animations, along with handling scaling of the input models so that the output structures are of a reliable size.

These structures are serialized via protocol buffers, allowing the serialized structures to be used cross platform, and to remove the requirement of handling serialization of arbitrary data.

With the core project meeting all success criteria, work has begun on the split rendering extension. The main goals for which are to evaluate the structures for use in a split-rendering environment (across a server and client). To begin, structure agnostic methods are being implemented, which includes image imposter, environment mapping, and depth image techniques. These methods provide an increasing amount of moveability from the client between received frames, at the cost of increased amounts of computation on the client side. Image Imposter involves sending only the rendered frame, environment mapping involves sending a 3D cube map of the environment, and sending the depth image allows for 3D image warping techniques to be implemented. Implementing these methods will allow for a qualitative comparison over the costs of the structures in a server, based on the resource requirement of the GPU and the bandwidth.

To facilitate the communication between the server and client, quic is being used as the networking protocol, to allow for both lossy and lossless communication. Messages are structured using protocol buffers, allowing for easy extensibility and reliable cross-platform communication. Individual image frames are encoded using ffmpeg which can then be sent over Datagram messages before being decoded on the client.