CSC2521 - COMPUTATIONAL DESIGN AND FABRICATION
FALL 2017

Assignment 1

Voxelization

Name: Lawson Fulton
Student ID: 998262062
Spetember 26, 2017

Results:

Basic Functionality :

I implemented a basic voxelizer using a single ray cast to determine membership. Results for 32x32x32 and 64x64x64 can be seen below for the provided meshes.

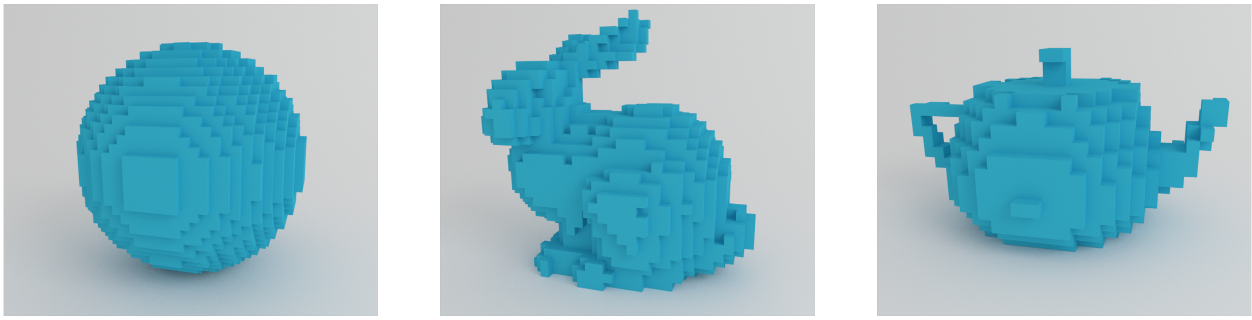

Figure 1: 34x34x34

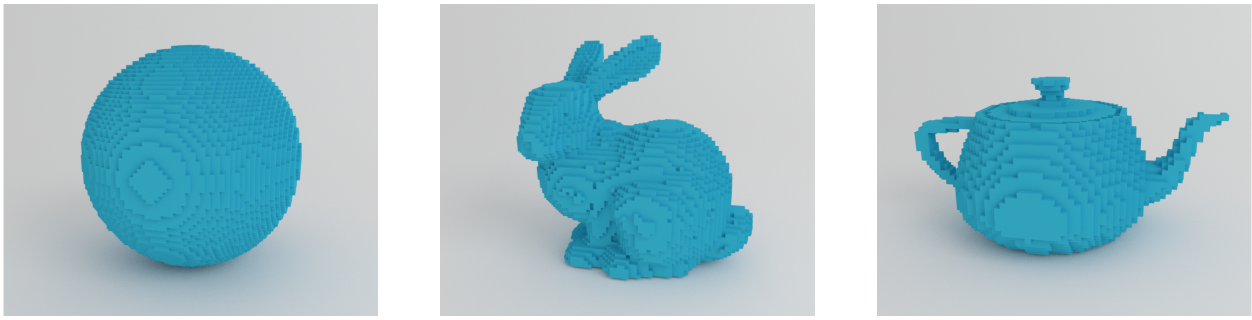

Figure 2: 64x64x64

Extra Functionality :

I also implemented a k-d tree to overcome the main bottleneck of intersecting the ray with every single triangle. This reduces the complexity from $O(mn^3)$ to $O(\log(m)n^3)$ where m is the number of triangles and n the grid resolution.

In figure 3 you can see a model initially containing 100,000 triangles voxelized at a resolution of 128x128x128 that completed in 31s.

Finally, I implemented an averaging of the result of multiple rays cast in different directions. This enables the voxelization of non-closed meshes, as you can see in figure 4.

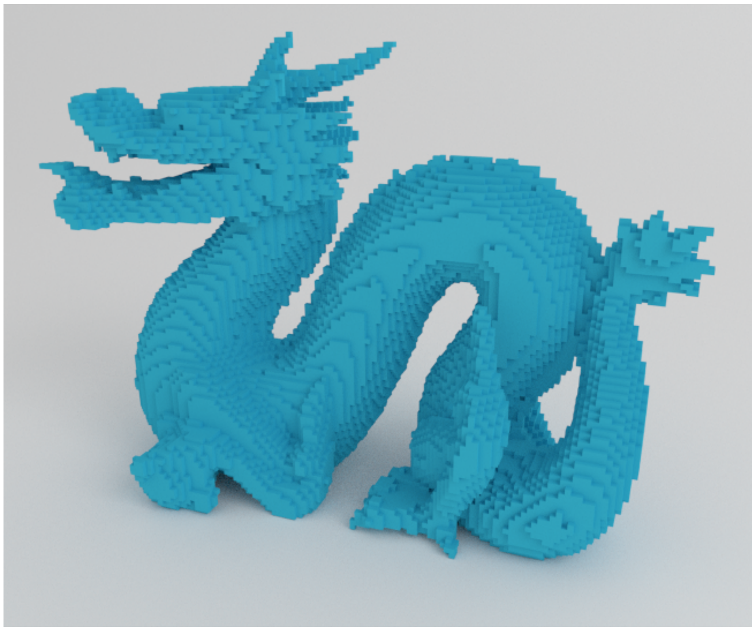

Figure 3: 128x128x128 Stanford Dragon

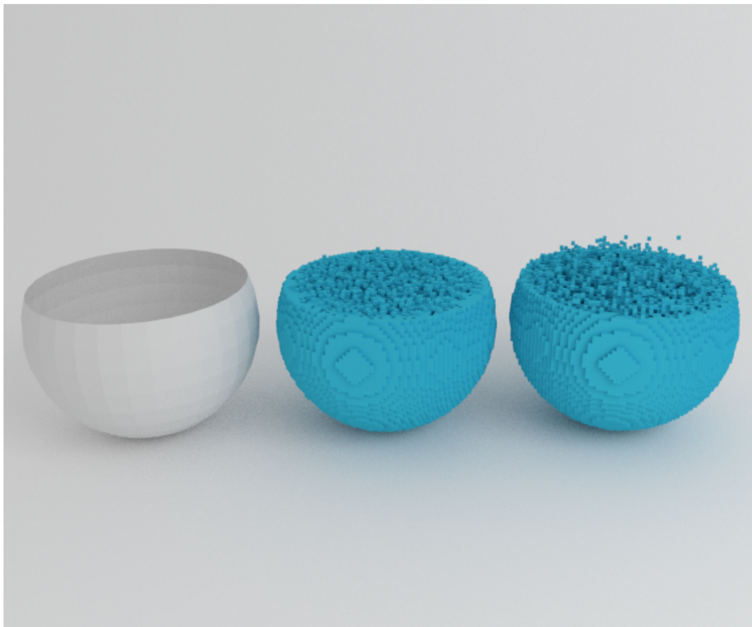

Figure 4: Left to right: Source mesh, 200 samples per voxel, 50 samples per voxel

Sources :

1. Fast algorithm for ray-triangle intersection: Moller-Trumbore intersection algorithm.

2. Branchless ray to bounding-box intersection algorithm: http://tavianator.com/2011/05/fast-branchless-raybounding-box-intersections/.

3. Stanford 3D scanning repository for Stanford dragon mesh: http://graphics.stanford.edu/data/3Dscanrep/.