Voxelizer Design Document

Stanley To

In this project, we implement a voxelizer! Our program takes as an input a triangulated surface mesh and the dimensions of the requested voxelization and outputs an appropriate voxelation. I built on top of an existing code base, implementing a function performing a ray-triangle intersection test (rayTriangleIntersection() in main.cpp), a function counting the number of intersections between a ray and a surface (numSurfaceIntersections() in main.cpp), and a function looping through each voxel and deciding whether it is inside or outside of the surface (main() in main.cpp). I referenced Computational Fabrication (CS581) lecture and lab material posted on Blackboard to inform my implementation. Visually, there does not appear to be any known problems in my code based on the produced voxelations. Below, I will exhibit the output results of three meshes under 32x32x32 resolution and 64x64x64 resolution.

Sphere

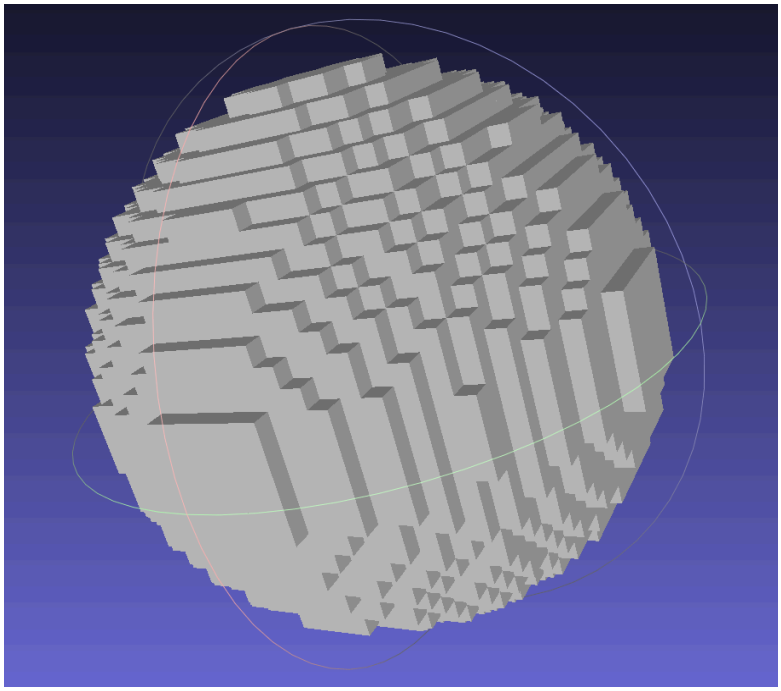

Figure One: A 32x32x32 resolution voxelation of a sphere (sphere.obj)

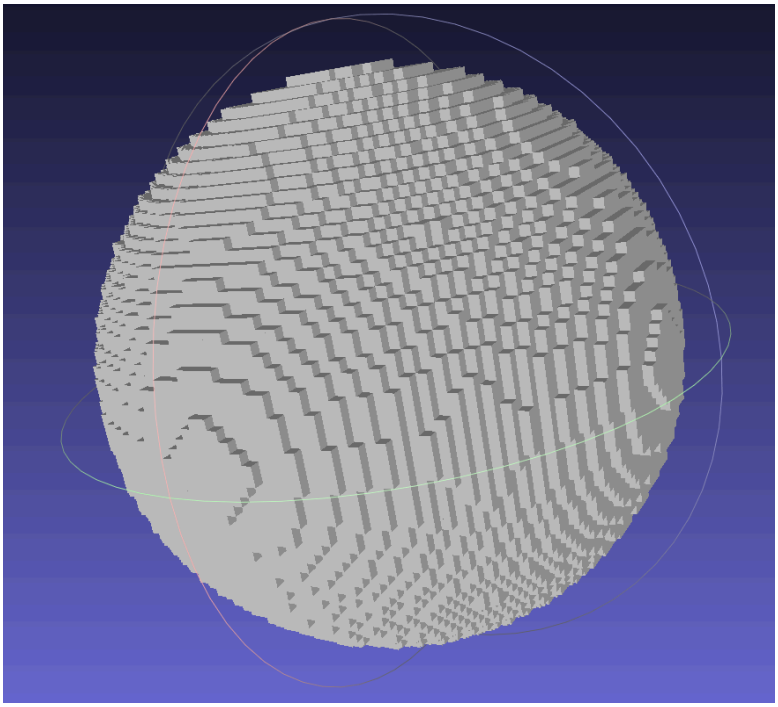

Figure Two: A 64x64x64 resolution voxelation of a sphere (sphere.obj)

Teapot

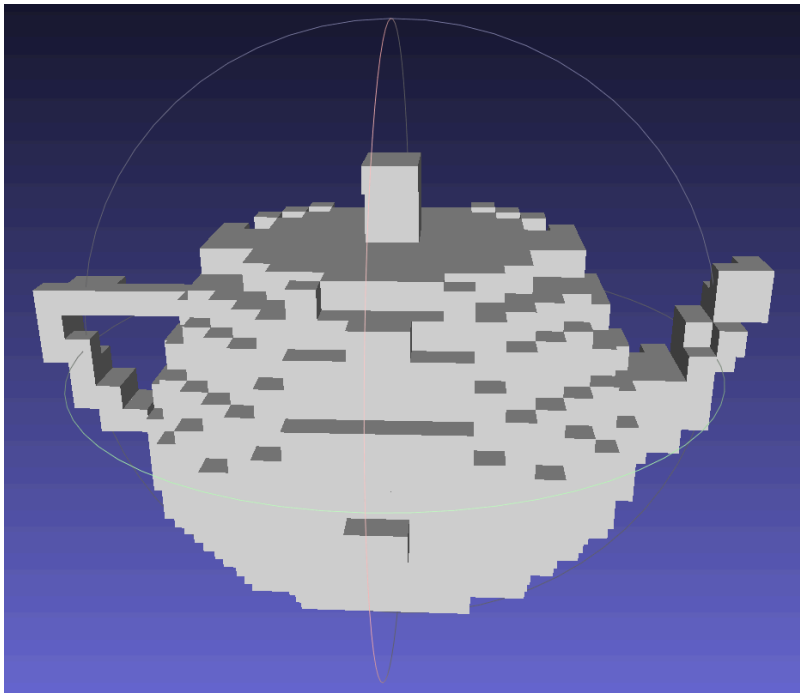

Figure Three: A 32x32x32 resolution voxelation of a teapot (teapot.obj)

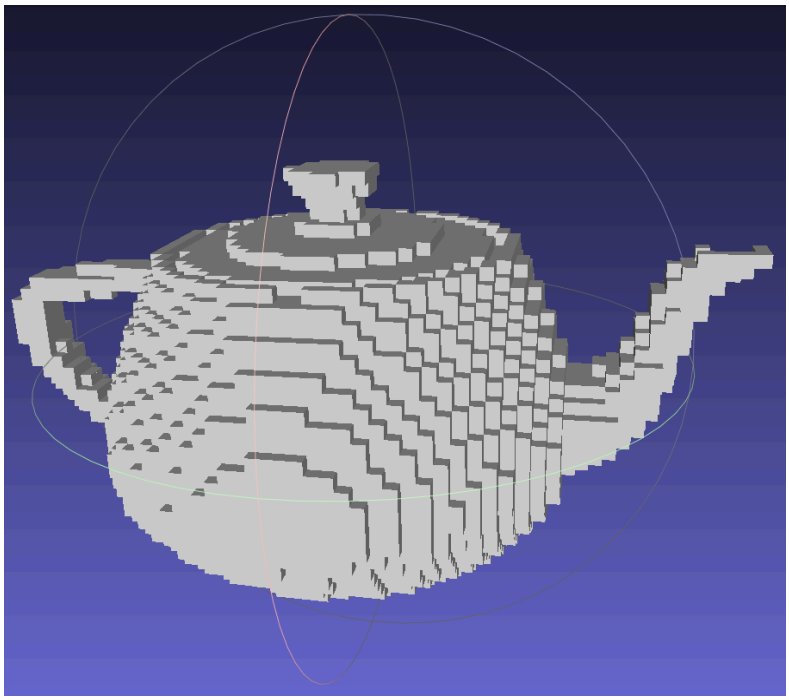

Figure Four: A 64x64x64 resolution voxelation of a teapot (teapot.obj)

Bunny

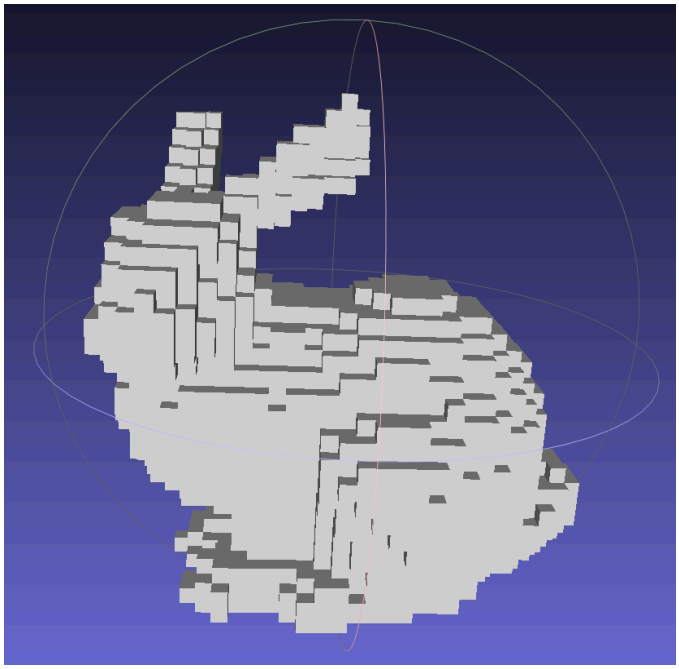

Figure Five: A 32x32x32 resolution voxelation of a bunny (bunny.obj)

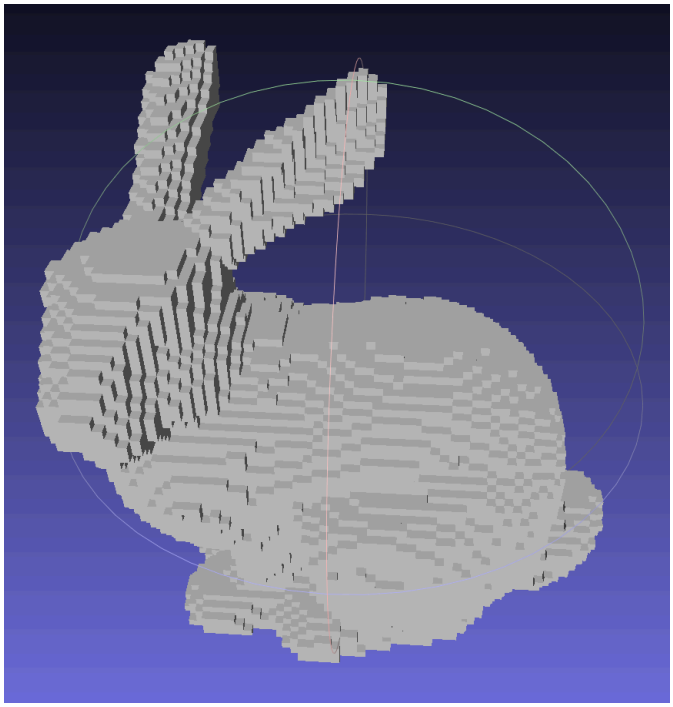

Figure Six: A 64x64x64 resolution voxelation of a bunny (bunny.obj)