

Milestone

Abstract -

To design a python based render engine that performs raytracing and output image files. Using multiprocessing library of python multiple images are rendered and the time taken is compared when rendering without multiprocessing.

Introduction -

Rendering engines have always intrigued me, in order to render a scene as close to reality means that human has understood everything about light and materials. This paper focuses on implementing a small render engine that takes shapes(object) as coordinates in a 3d space, defines material properties to it applies light interaction on it to render the output.

Background -

Objects in the scene are defined by polygons, this is called mesh and the quality of the render highly depends on the details of the mesh, in many applications a high polygon mesh is compressed to low polygon mesh to decrease the processing time taken. For the scope of this project, objects are defined by coordinates.

Methodology -

Used volume ray casting then applied raytracing to get the RGB values onto the defined screen. These RGB values are then used in creating a png file.

Experiments/Analysis -

The core render program was then tested using concurrent programming with different inputs. Using multiprocessing library has shown 50% increase in efficiency. The efficiency still majorly depends on the number of objects and the distance from camera.

References -

Volume ray casting -

[Volume Ray Casting part-1](#)

[Volume Ray casting part-2](#)

Ray Tracing -

[Ray Tracing](#)

Implementation Reference

[GIT HUB](#)