Raytracing Project

Jared Givens, Ahbi Sohal, Noah Krim

28 May, 2023

Serial Implementation The program outputs color values of pixels with raytracing techniques. It begins by initalizing virtual geometry and a camera. Then it sends rays from the camera into the virtual space.

For each pixel a series of uniformly random rays are cast to gather a color value. the color values are then averaged to achive anti aliassing.

Each ray iterates over the scenes geometry to determine the nearest intersection. The nearest intersection's color is then multiplied against the rays running total color value. The ray then reflects or refracts based on the properties of the material that was intersected. The process of accumulating color and bouncing continues until the ray fails to collide with an object or reaches the maximum ray depth. If the ray fails to collide, the program adds the contribution of the skybox to complete the color. If the ray reaches the maximum bounce depth the color is set to black.

The serial prototype is only capable of outputing frames after several seconds. This program is an excellent candidate for CUDA because the code of each ray can be run in parallel. Additionally the code for each ray contains many branches that could be removed with further linear algebra.

Our team can complete this project because we completed a serial prototype in cpp this weekend:

Repo

Following the tutorial:

RayTracingInOneWeekend