

Minecraft Hardware Accelerator (Ray Tracing)

winixent + yohang

Project Abstract

18 October 2024



Everyone's favorite voxel game, *Minecraft*, now supports ray tracing! Even more beautiful than the most high-fidelity of shaders, but also dependent on modern (and expensive) GPUs — it's out of reach for most of us! This is because ray tracing, unlike rasterization, is an inherently computationally-intensive task where the trajectory of light is simulated per pixel (of which there are many), each frame. We propose a *Minecraft Hardware Accelerator* which replaces the vanilla rendering pipeline using an FPGA raytracer. Using a mod, we plan to stream world (voxel) and player position data to an Urbana board which will re-render the world with these new, nicer graphics. The FPGA will then display the framebuffer to a monitor over HDMI, as we've done in class.

Some goals for this project:

- Ray-tracing on an FPGA
 - We target axis-aligned voxels, which can be traversed faster than meshes (even with BVH).
 - Experiment with sparse voxel octrees; can we fit large chunks of the world in memory despite the Urbana board's modest resources? Would traversing an octree be faster than a naive 3D array?
 - Can we pipeline the per-pixel process, then instantiate it many times over to compute each frame in parallel?
- Minecraft Mod/Plug-in
 - Written in Java, this is a small (but necessary) part of the project.
 - Streams the player (camera)'s position and world data each frame.
 - User input/gameplay is the same as usual; our implementation is opaque to the Minecraft client.
- *Stretch goal*: Multi-FPGA Accelerator
 - Since ray-tracing is inherently parallel, and parts of the framebuffer can be computed independently, can we employ multiple FPGAs to render to the same display?
 - If it happens that one FPGA is enough for >30FPS, we could increase screen resolution or light bounces.