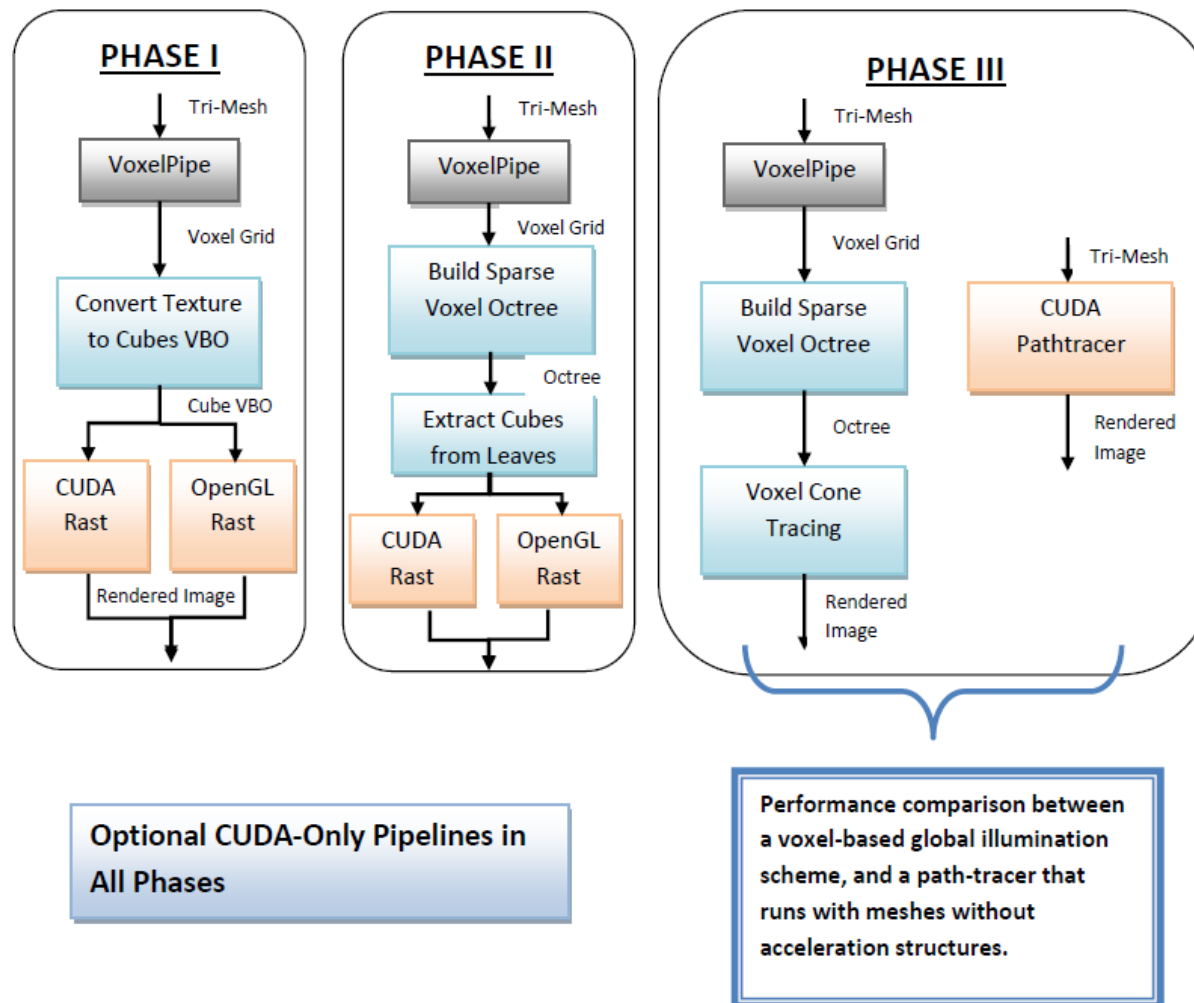


A Voxel Rendering Pipeline in CUDA for Real-time Indirect Illumination

Dave Kotfis

Jiawei Wang

Project Proposal



Tasks

Complete

- Project Software Configuration
- Literature Review
- OpenGL/CUDA Rasterizer Setup
- VoxelPipe Software Integration
- Cube Extraction from Voxel Grid

TO-DO

- Voxelization Improvements
 - Geometry Shader
 - Optimizations
 - Texture Mapping
- Sparse Voxel Octree
- Voxel Cone Tracing
- Performance Evaluation

VoxelPipe

- Coarse Raster
 - Parallel on Triangles
 - Bucket into Tiles
 - Create fragments (triangle, tile)
- Radix Sort On Tiles
- Fine Raster
 - Compute Voxels
 - Blend Over Triangles
- New Steps:
 - Encode voxel centers (occupied only)
 - Stream compaction
 - Copy cube VBO at each center

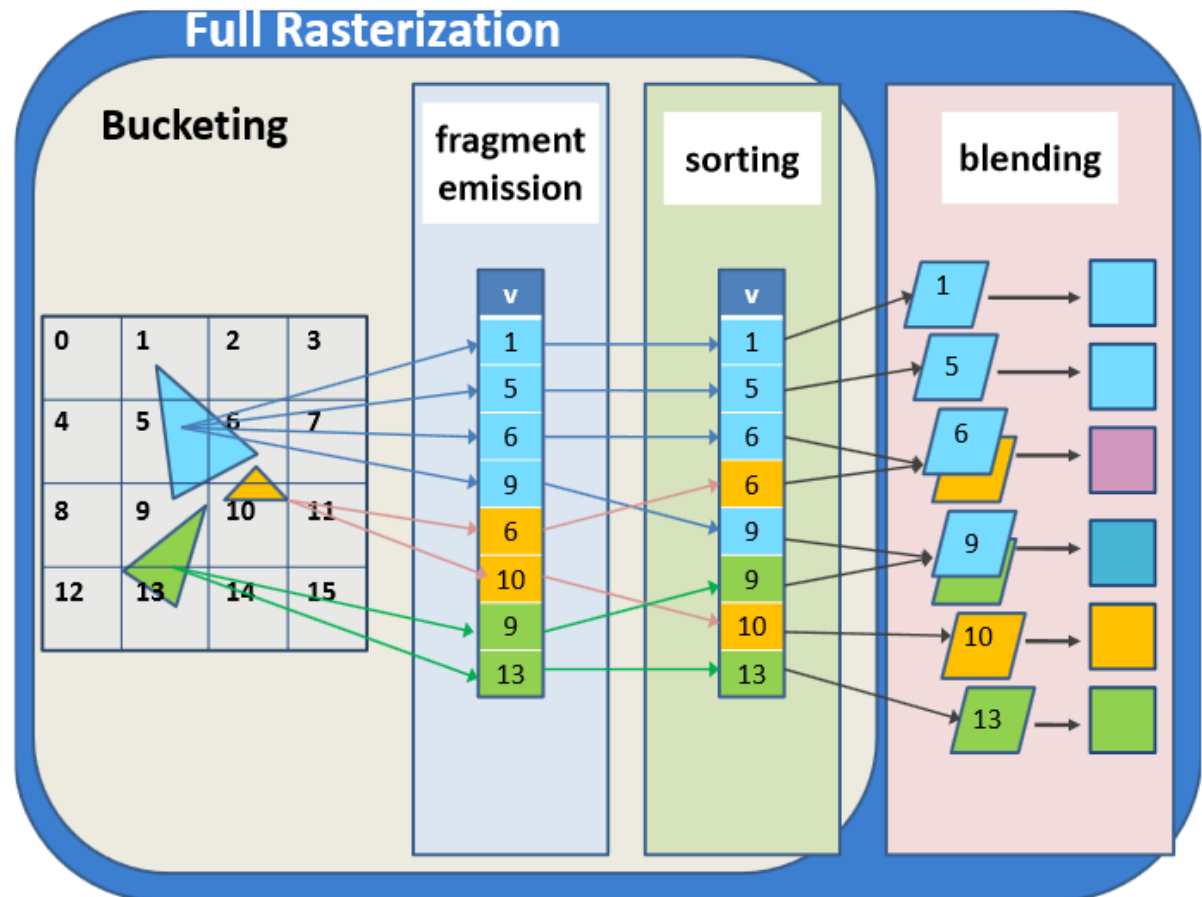
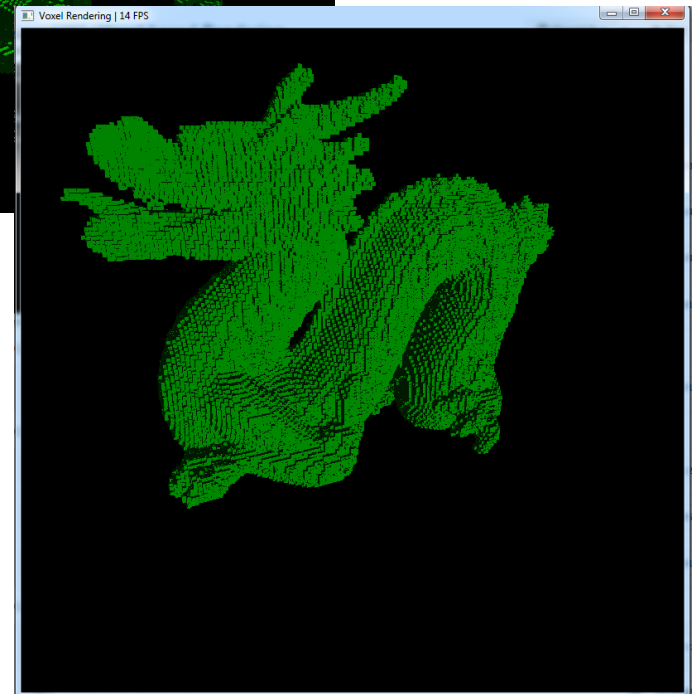
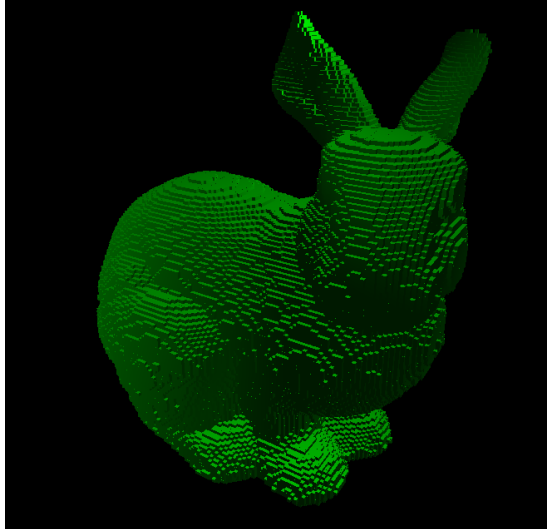
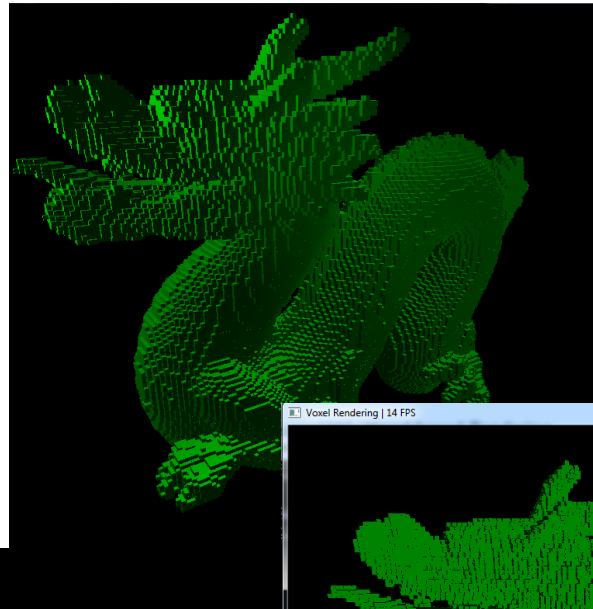
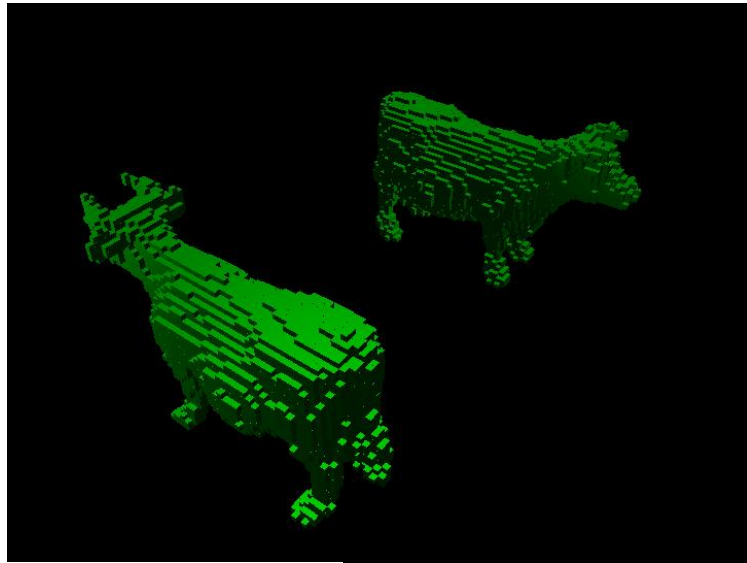


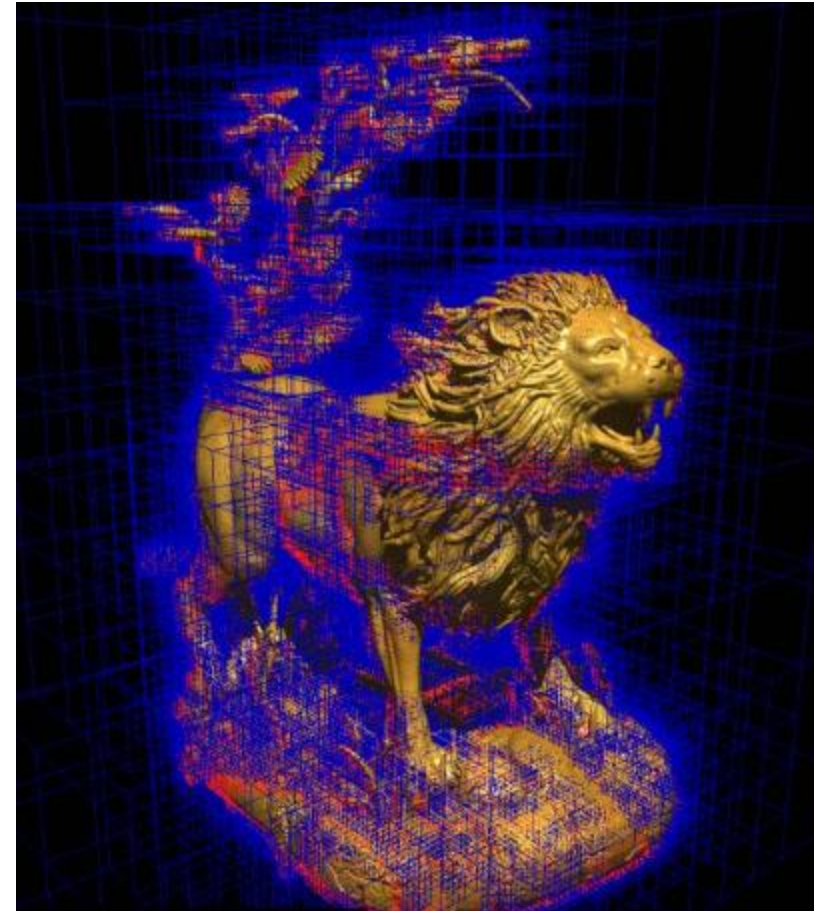
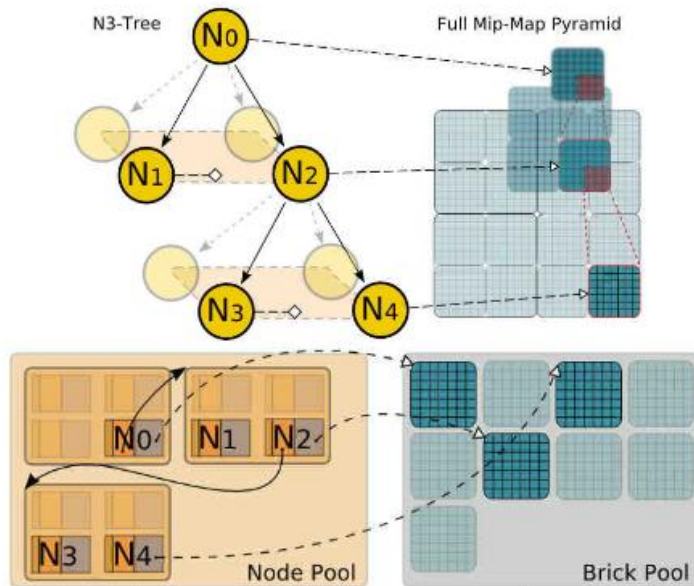
Image Credit: Jacopo Pantaleoni, NVidia Research

Results So Far..



GPU Sparse Voxel Octree

- Node pool in GPU memory.
- Brick pool in texture memory for interpolation.



Voxel Cone Tracing

- Use Octree LoD for efficient cone integration
 - Avoid slow Monte Carlo from our CUDA Pathtracer.
- Primary Rays -> Normal Rasterization

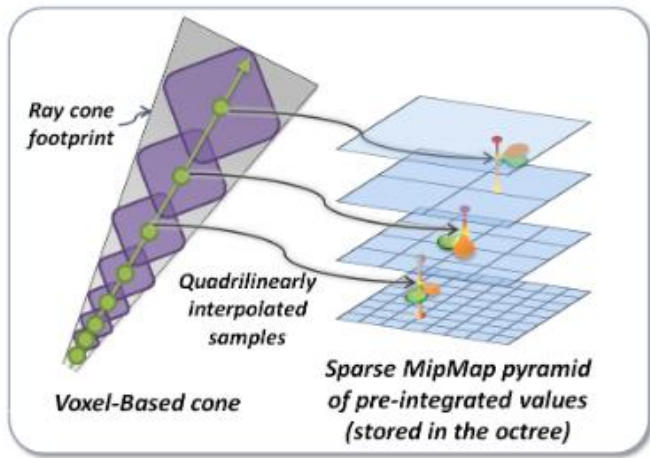


Image Credit: Cyril Crassin