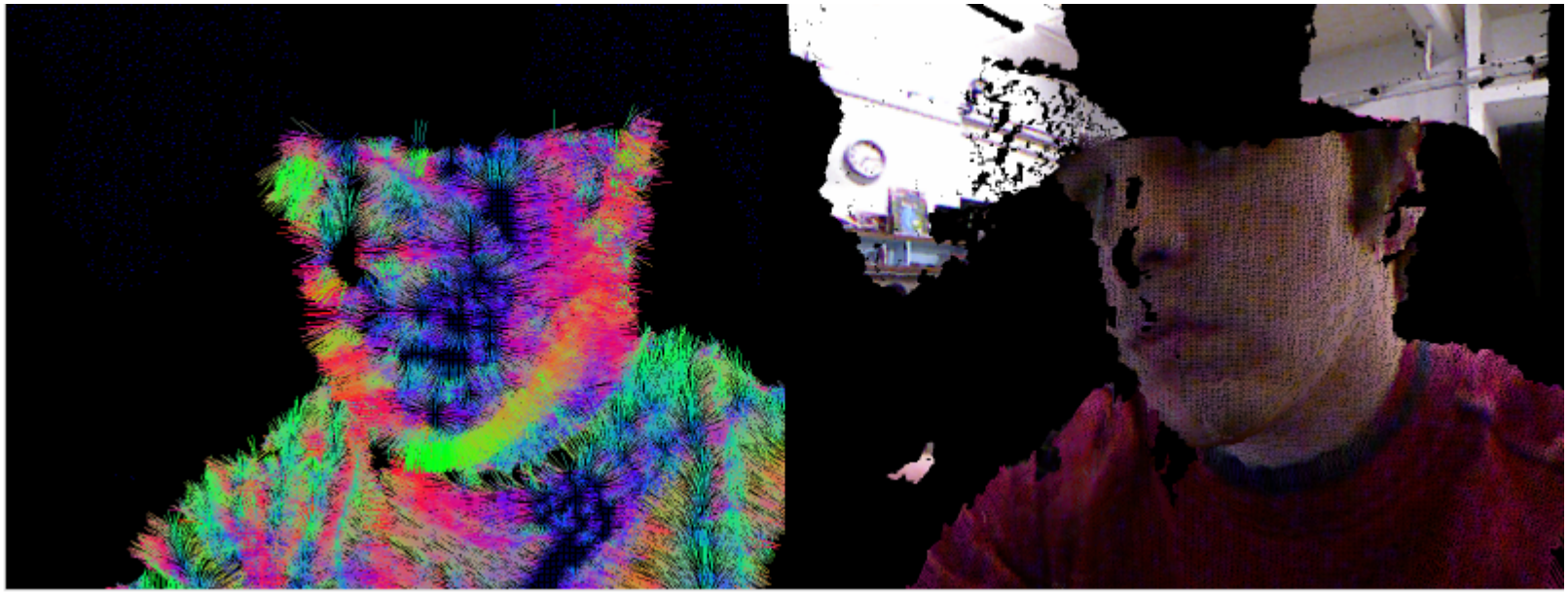
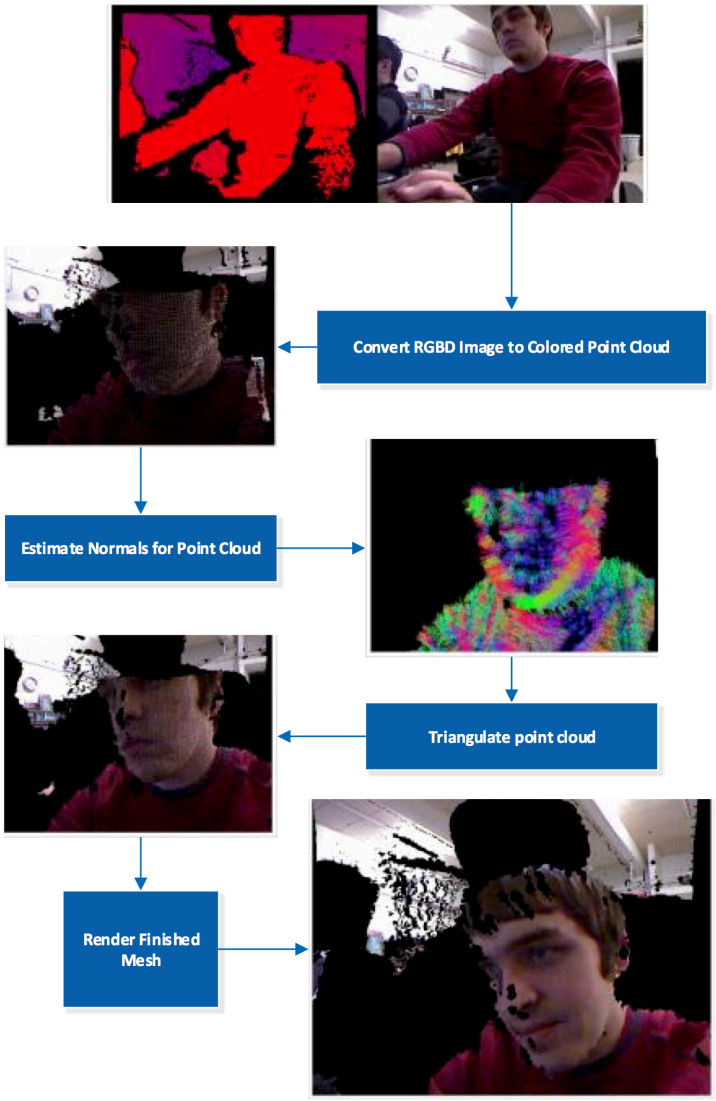
GPU-Accelerated Conversion of RGBD Images to Textured Triangle Meshes



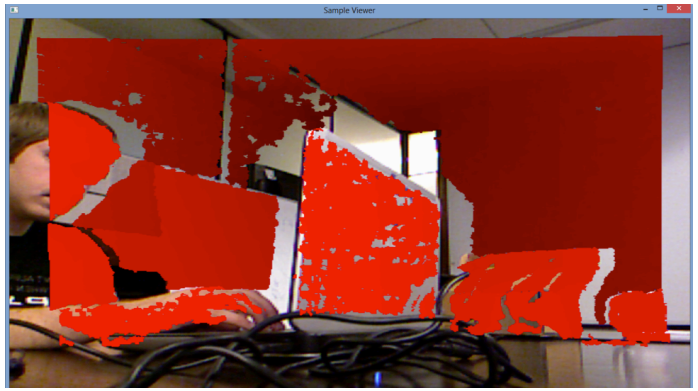
Collin Boots and Dalton Banks

Image Processing Pipeline

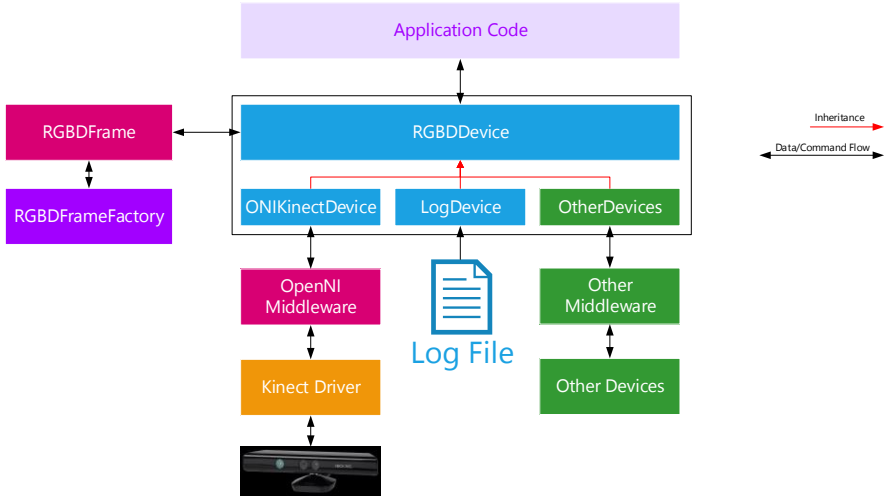


Framework

- Capture, synchronize, and optionally save RGB and depth images streaming from RGB-D capture device
- Display and process either live or recorded RGB-D streams
- All processing and rendering doing on the GPU



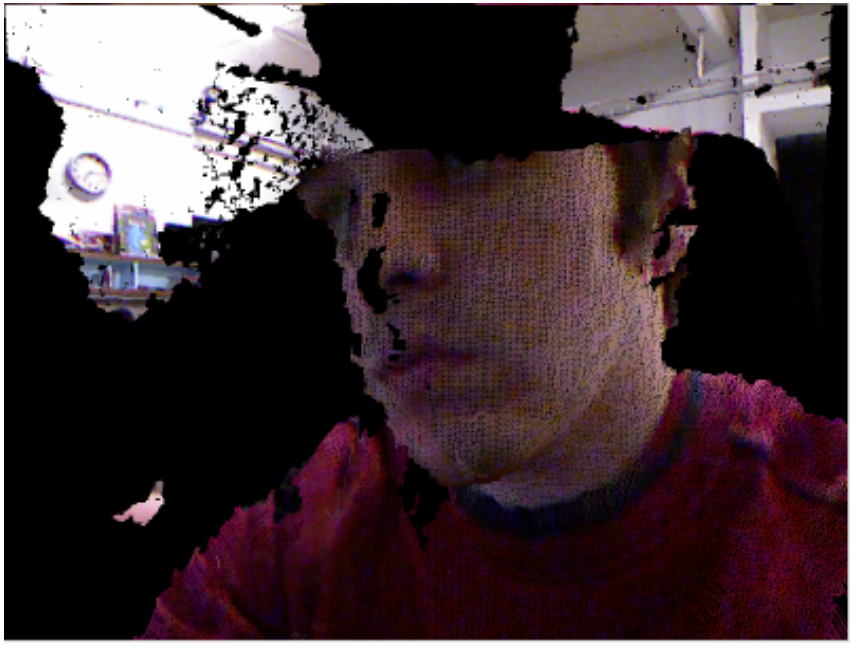
Synchronized RGB and depth streams



Framework architecture

Point Cloud Reconstruction

RGB

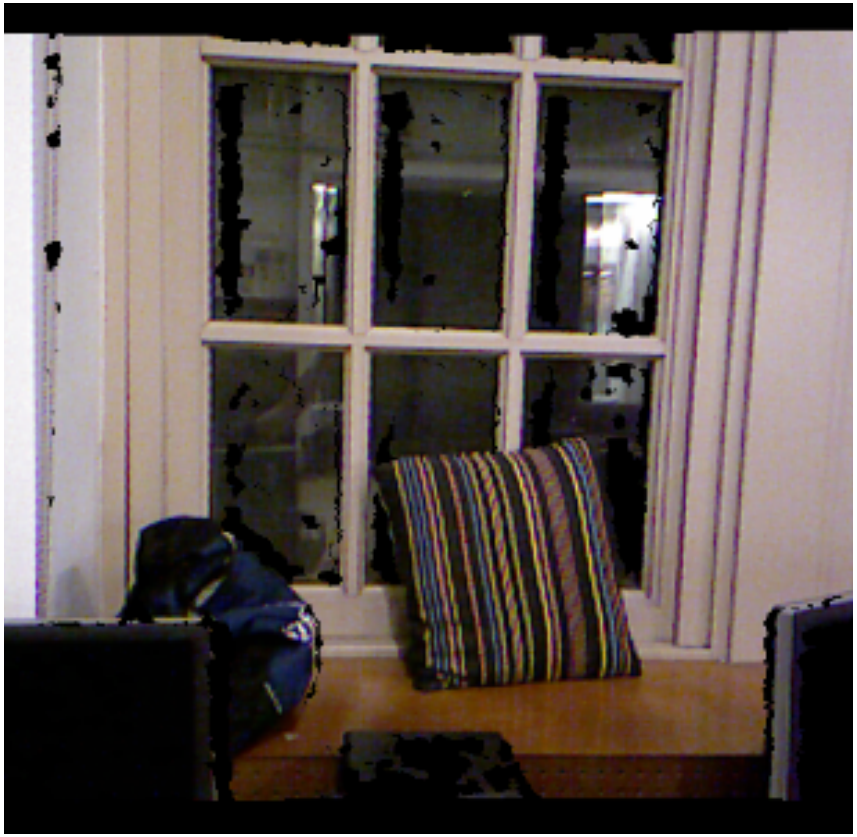


point cloud

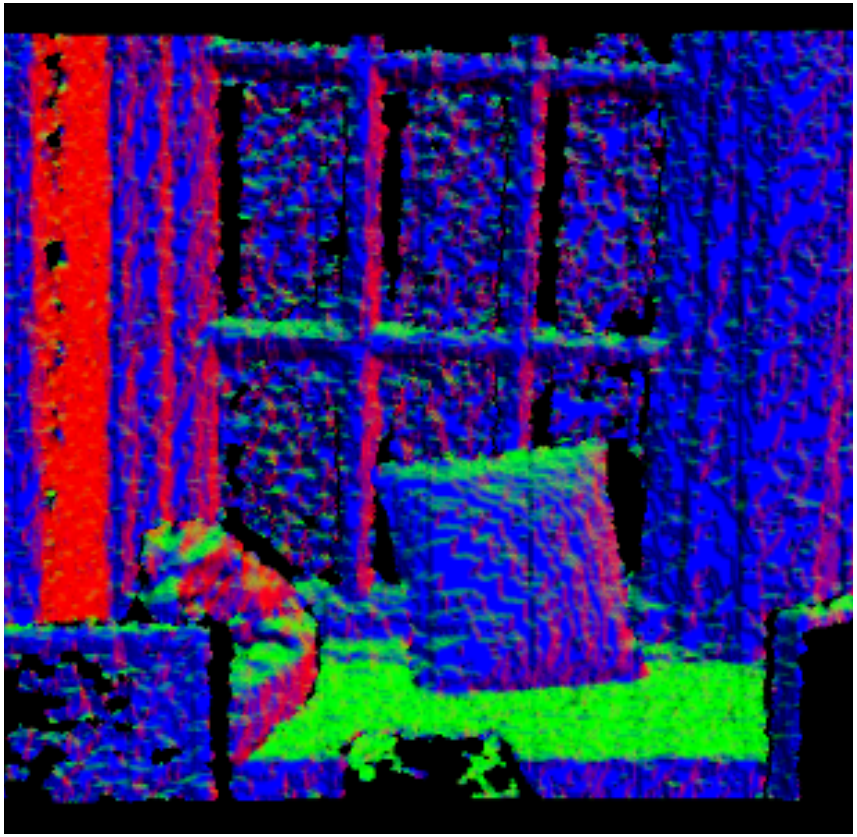
Point Normals Estimation

- Identify nearest neighbors
  - First pass: XY image window
  - Second pass: 3D radius threshold
- ~~Estimate normal from NN best-fit plane using PCA~~
  - Too slow: reimplemented using averaging
- Assign direction using dot-product with camera vector

Point Normals Estimation

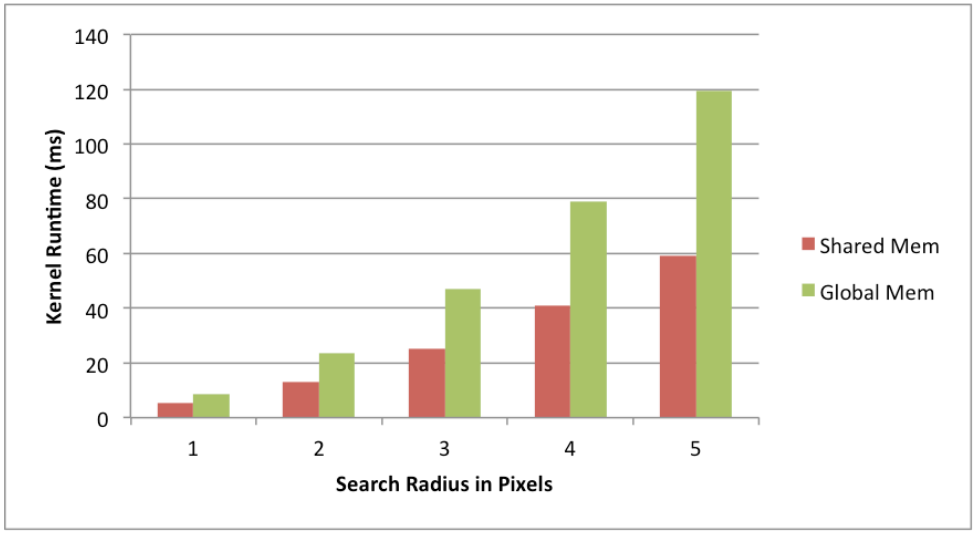


RGB

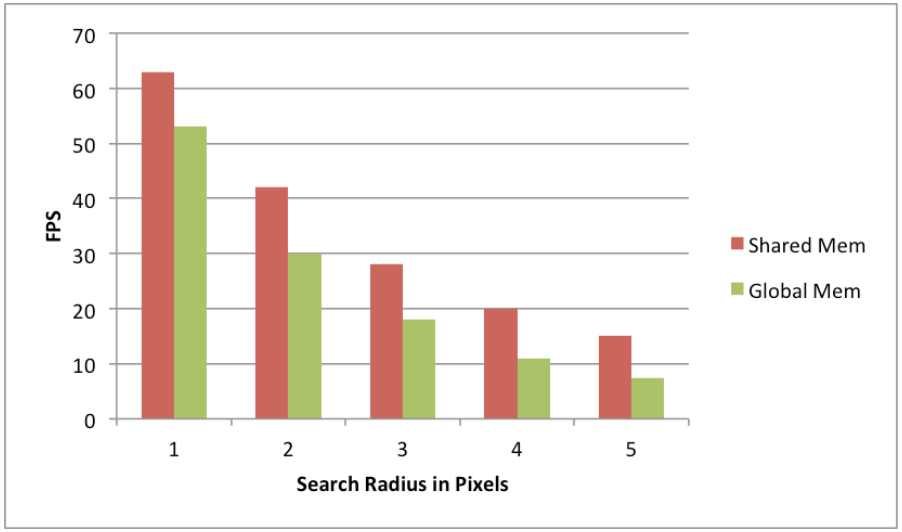


point normals

Performance Optimization



kernel execution time



frame rate (fps)

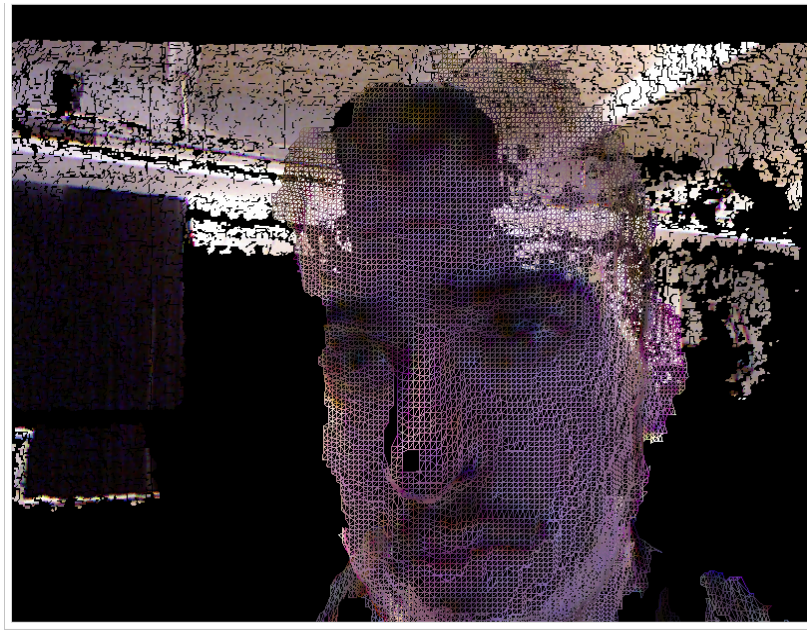
Mesh Reconstruction

- High resolution mesh (all valid vertices)
- Parameterize by triangle

Mesh Reconstruction

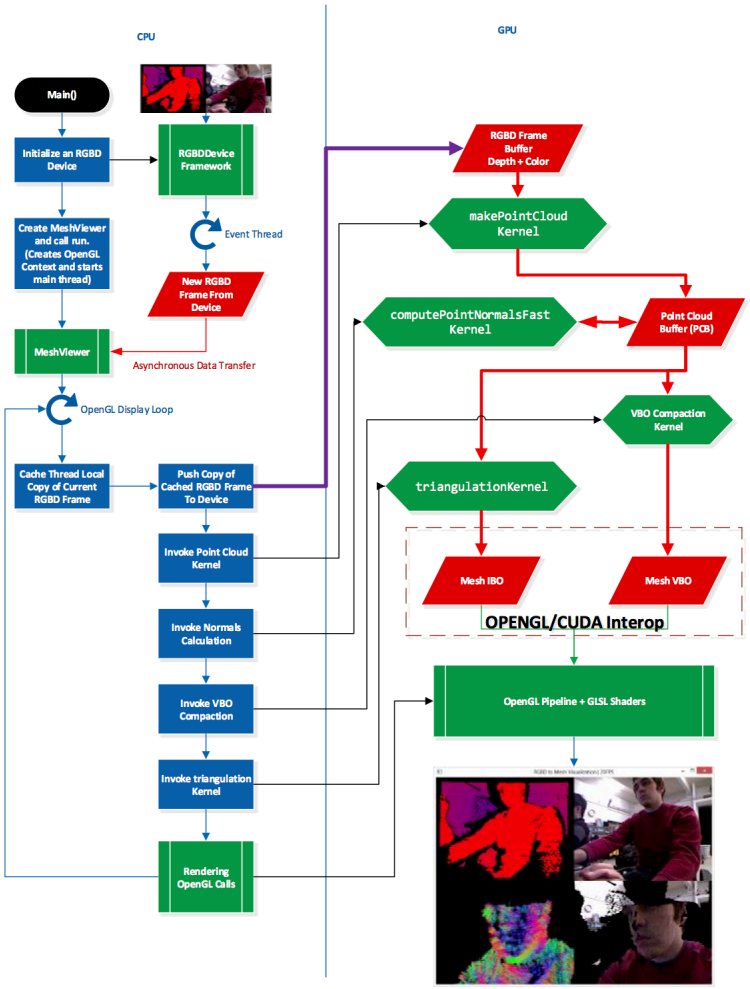
- High resolution mesh (all valid vertices)
- Parameterize by triangle

filled

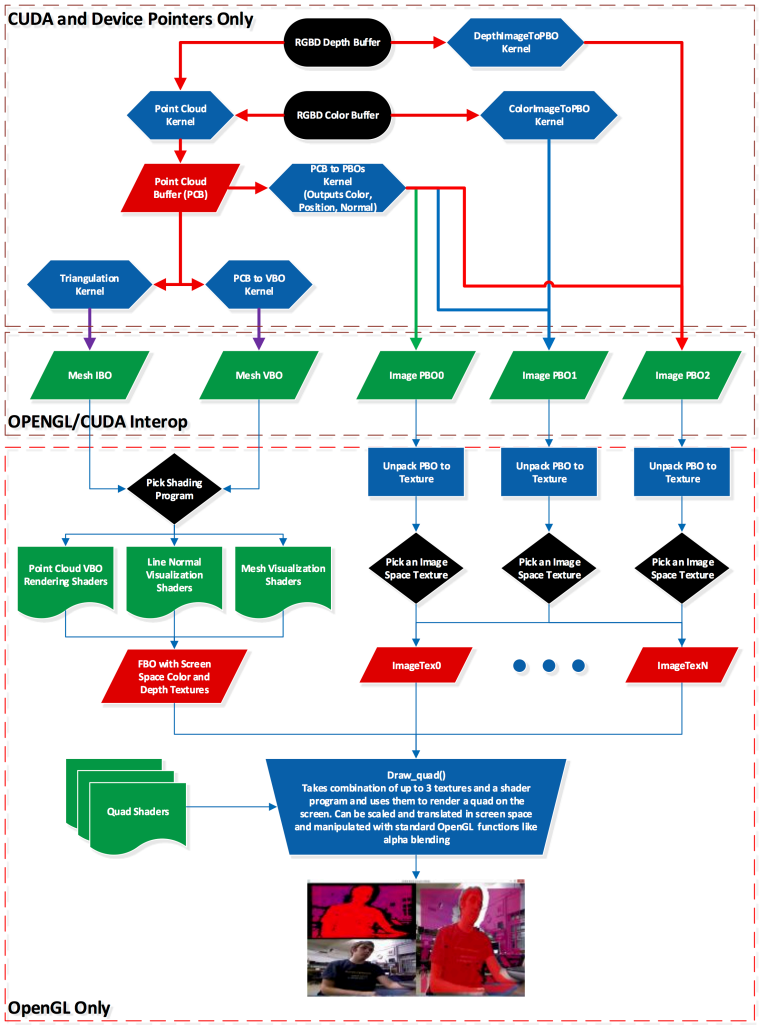


wire frame

Full Pipeline / Interactive Shaders



program overview



rendering pipeline

Live Demo!

Future

- Determine if improved normals are necessary for accurate surface reconstruction
- Account for Kinect depth digitization
- Adaptive mesh simplification
- Poisson surface reconstruction?
- Image registration and mesh merging