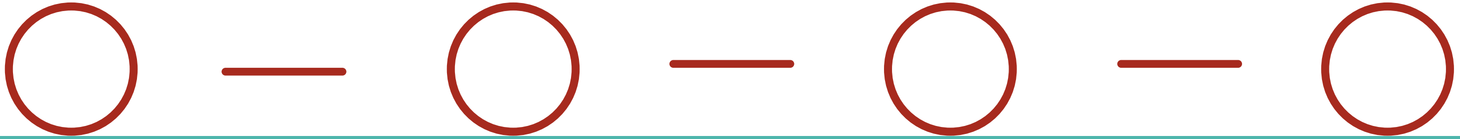
CUDA Based 3D Descriptor for Object Detection

Yu Sun • 12.10.2018

M.S. Robotics

3D Object Detection Pipeline



Downsample the point cloud (features)

Compute Normals

Compute Descriptor

Local reference frame

Use normals and color information

Descriptor Correspondence

Nearest Neighbor

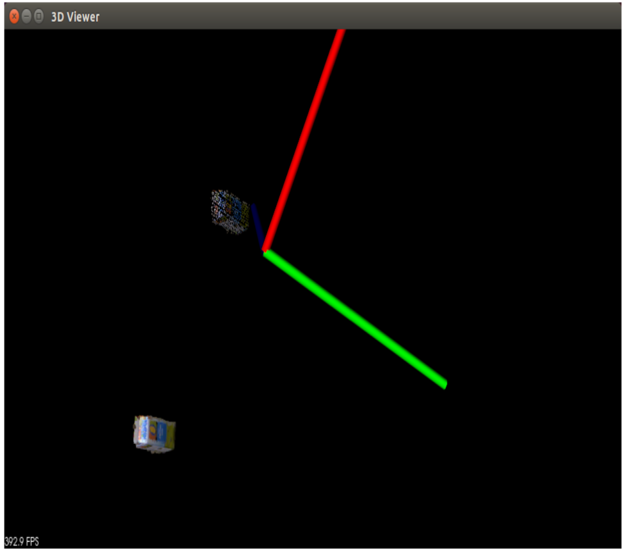
Grouping

Consistency

Point Cloud Downsampling

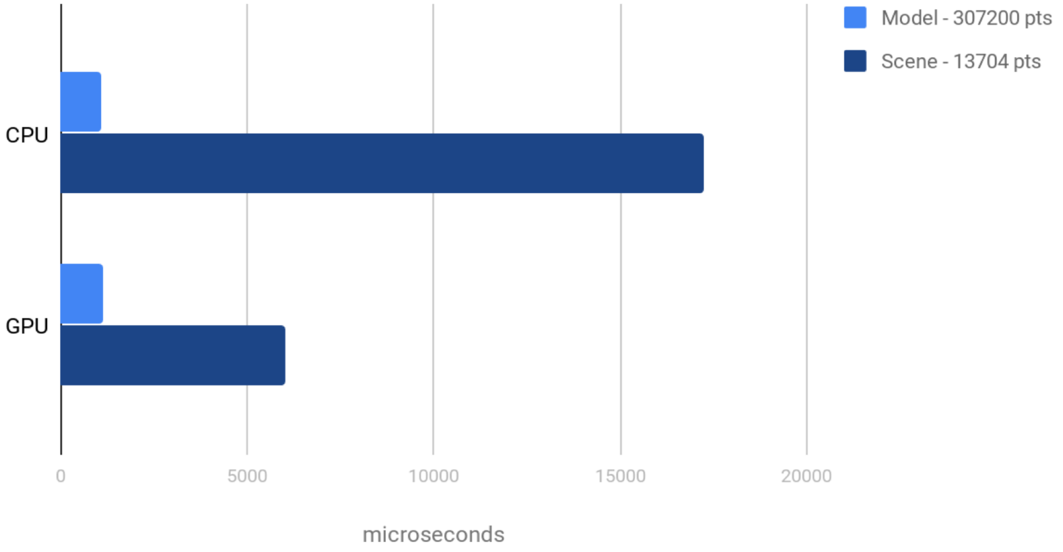
Save Memory and Computation / Use as Features

- To best keep the original shape of the point cloud, instead of uniform sampling of points, divide the point cloud into multiple grids and keep only the point closest to the center of the grid it belongs to.
- Choice of resolution requires experiments. Bad choice could lead to completely incorrect detection result.



Performance Gain (Downsampling)

Uniform DownSampling Performance Analysis (Lower is Better)



CPU	1085 μs	17216 μs
GPU	1129 μs	6021 μs

Descriptor SHOT

Color-SHOT / SHOT

- Chain signatures of histogram relative to different <property, measurement> pairs
- Color and Normals
- RGB to CIELAB color space (abs difference)
- Dot product for normals (local reference frame · every neighbor within radius)
- Interpolate single/both channels’ histogram

F. Tombari, S. Salti, L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching", *IEEE International Conference on Image Processing (ICIP)*, September 11-14, Brussels, Belgium, 2011.

SHOT (Local Reference)

Refined Normal Estimation

- Weighted Total Least Square **M** (Distant from Feature)
- Eigenvalue Decomposition of covariance matrix **M**
- Covariance Matrix calculated using neighbors within pre-set radius

Disambiguation of directions

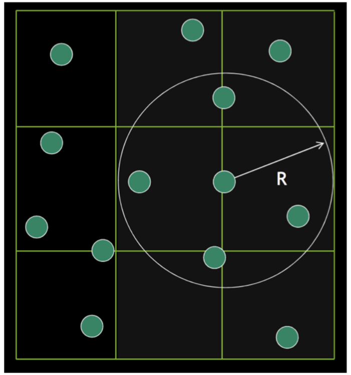
- Eigenvectors with decreasing eigenvalues
- Choose sign of eigenvectors coherent with majority of vectors it’s representing

F. Tombari *, S. Salti *, L. Di Stefano, "Unique Signatures of Histograms for Local Surface Description", *11th European Conference on Computer Vision (ECCV)*, September 5-11, Hersonissos, Greece, 2010.

Radius Search

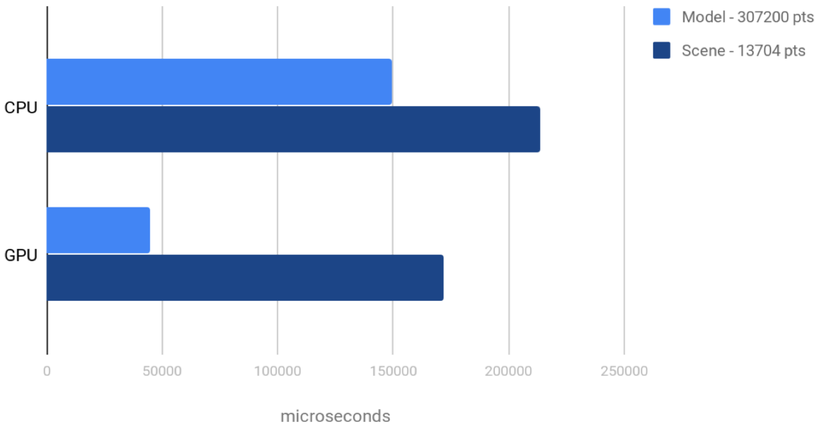
Normals, Local Information

- Instead of searching all candidates, split point cloud into grids.
- Identify which points are in which cell
- Only search the neighboring grids to find candidates
- To reuse computation from downsampling, the point clouds are not splitted into equal distance bins. Instead, I ensure each dimension has same number of bins.

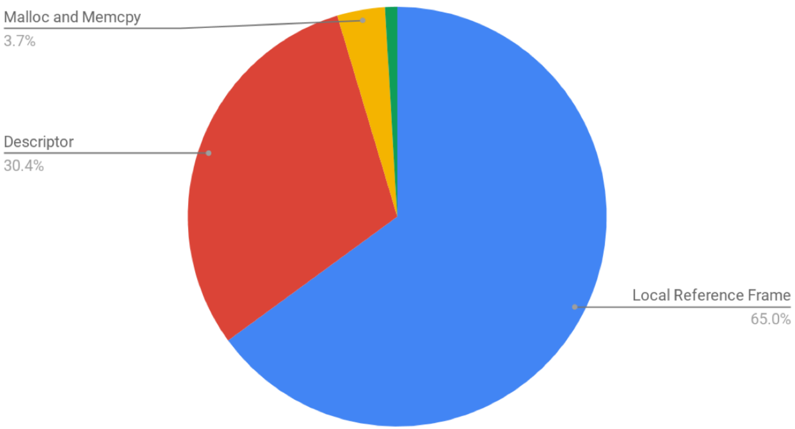


http://on-demand.gputechconf.com/gtc/2014/presentations/S4117-fast-fixed-radius-nearest-neighbor-gpu.pdf

Performance Gain (SHOT)



Descriptor Computation Performance Analysis (Lower is Better)



Computation Cost for Scene Point Clouds

CPU	149230 µs	213478 µs
GPU	44788 µs	171608 µs

Correspondence

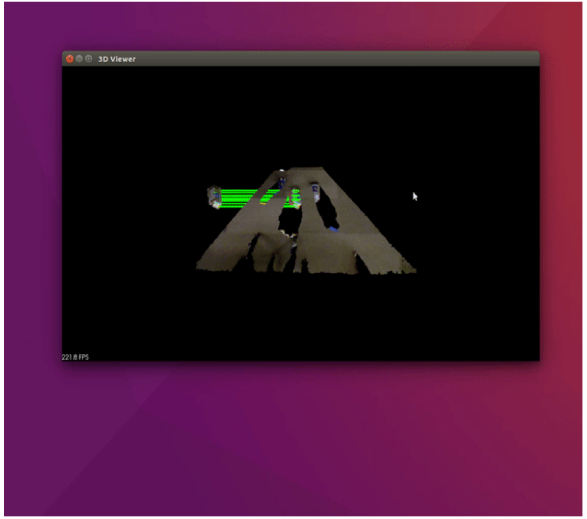
Nearest Neighbor

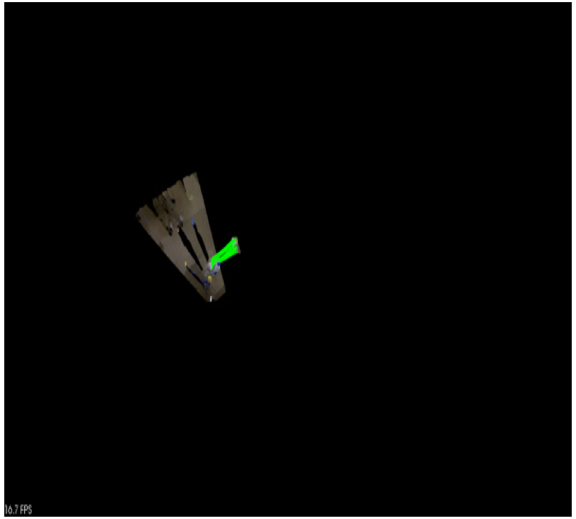
- L2 squared distance (within threshold)
- KDTree with user defined data dimension
- Brute Force Search

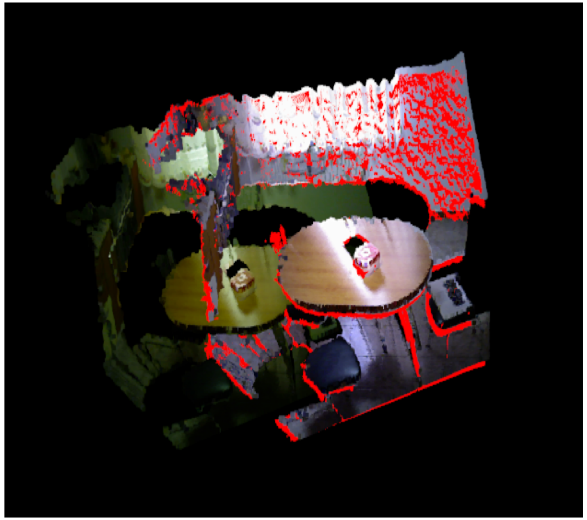
Performance

- KDTree and Brute Force takes about same time for real searching computation, but KDTree forming needs extra time on CPU and extra memory
- 94 % time writing to output
- 14 X faster than total CPU time

Demo (~ 270 fps without display)







Reference & Last Thoughts

PCL Tutorial and Documentation

Eigen Tutorial and Documentation

Y. Guo, M. Bennamoun, F. A. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3D local feature descriptors. IJCV, 116(1), 2016.