SEGMENTATION OF 3D LIDAR POINT CLOUD DATA
CSE-548 COMPUTER GRAPHICS
ANISH SAHA - 112027821

INTRODUCTION

The advent of LIDAR sensors has led to the generation of large volumes of point cloud data. Due to certain advantages such as depth information and reduced data volumes, they are highly beneficial in roles which otherwise image data would have fulfilled. This is a particular case of street view LIDAR point clouds that require segmentation into roads, buildings and miscellaneous clusters. A rule based segmentation first classifies about 75% of the points into road or building, Then a super-voxel generation module clusters, merges and classifies super-voxels. We shall look at each step sequentially.

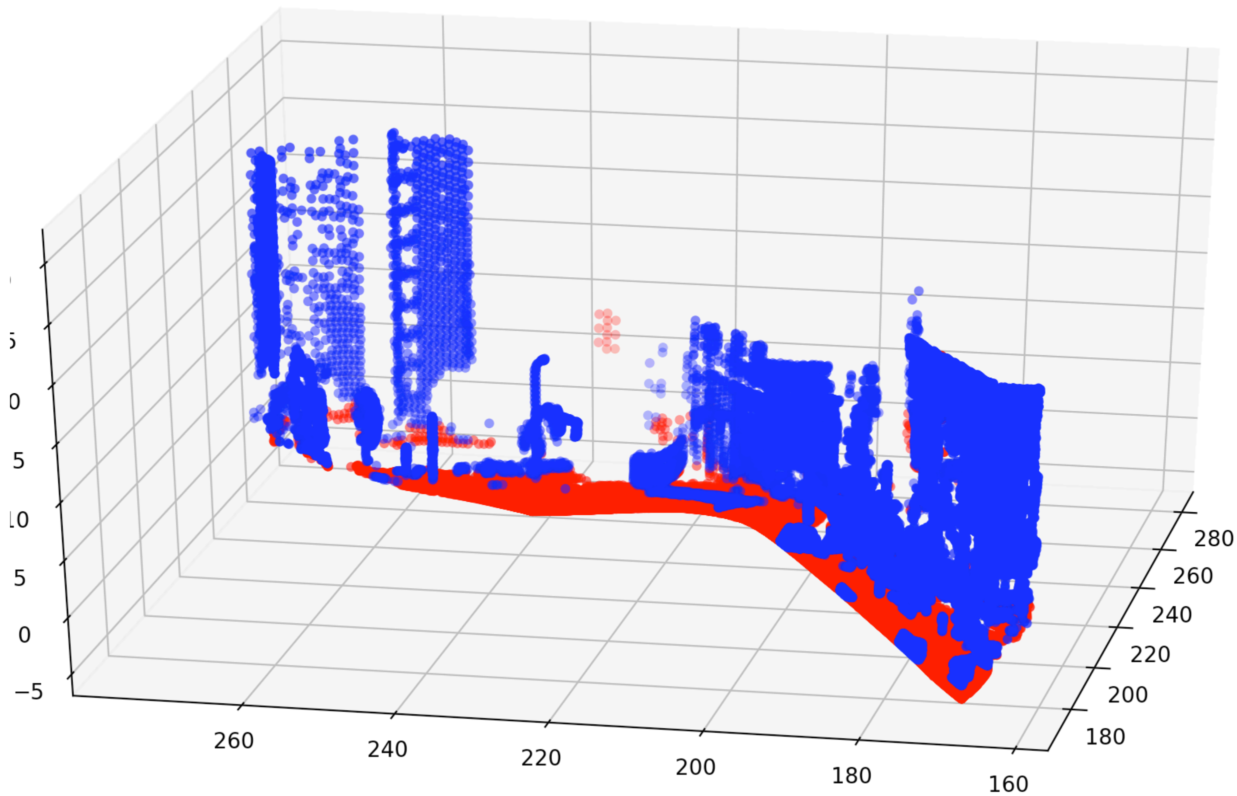
ROAD/GROUND SEGMENTATION

The entire scene is divided into equal side square blocks in the x-y plane. Each block then uses a RANSAC based fitting to try and fit planes to points with the lowest Z-values. The points with the 10-lowest values are averaged to create a minimum Z-values. All points within a certain threshold of the average are chosen for the RANSAC plane fitting. After much experimentation with dynamic iteration count for the RANSAC fitting, a count of15 was chosen to be most efficient. For any block, if the number of chosen points exceeds 7, a dynamic iteration count greater than 50 would be chosen which is highly inefficient as stated in the half way progress report. This caused a performance drop of more than 4 times. Hence, an iteration count of 15 was fixed for correct but efficient runtime.

Further performance improvements suggested are:

- Execute RANSAC on each block on parallel threads. Since there are no dependencies in plane fitting for each block, they can be executed in parallel. As we will see in the performance report, nearly 75% of the total runtime is expended in this module. Hence, a boost in runtime would increase overall performance significantly.
- Dynamic and adaptive block sizes for real time purposes are possible. The whole concept of blocks are required to accommodate for gradient in roads since, in areas with varying scene gradient, a single ground plane may not be sufficient to capture the entire ground. Thus block size can be increase in scenes with low variation in scene gradient. An increase in 2 times of the block length would reduce the number of blocks to $1/4^{th}$. Since, number of iterations is fixed, this could result in a performance improvement by 4 times.

ROAD/GROUND SEGMENTATION RESULT



BUILDING FACADE SEGMENTATION

For the building facade segmentation, the remainder of the points are again divided into blocks along the x-y plane. The height of each block is then compared against the global maximum height(Normalized) amongst all blocks. This ensures blocks with tall features are candidates for buildings. Notice that this also picks trees and street lights as possible candidates. The density score helps in removing candidates with small point cloud densities such as street lights or telephone poles. Now. the horizontal spread can be used to differentiate between the remaining building facades and trees. The ratio of the length of major axis and the ground area covered is a good measure to separate the two.

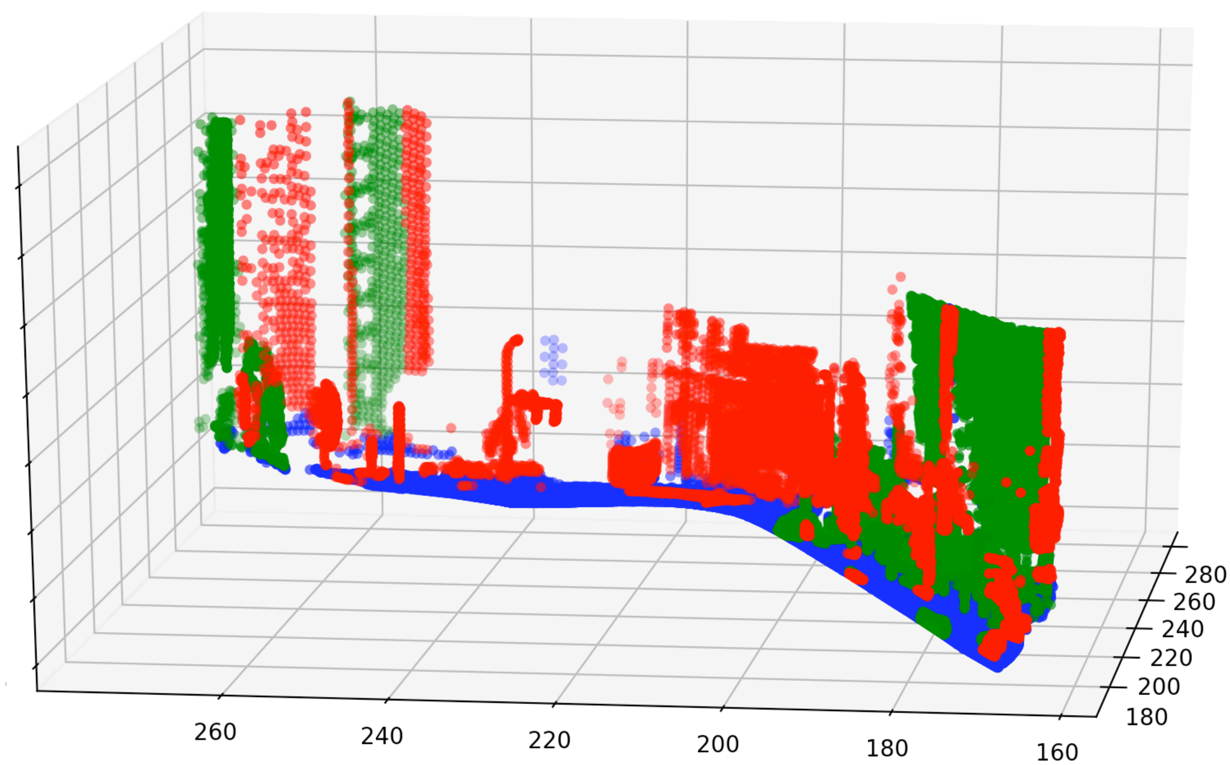
Improvements implemented for performance boost:

- For calculation of ground area of a point cloud, a bounding box algorithm has been ditched for a area of a bounding ellipse approach. The covariance of the data in the x-y plane is determined and Eigen vector is used to align the point cloud to the coordinate axes. The major and minor axes lengths are determined and the area of a bounding ellipse gives the ground area by the formula: Area of ellipse $= \pi * a * b$, where a and b are the major and minor axes lengths.

There are certain shortcomings noticed during the implementation of various cases:

- When the plane of a building facade is nearly parallel to the direction of view, a sparse number of points are reported. This makes the facade fail the density score. A possible solution is to carefully tweak to weightage of the density in the final building score calculation.

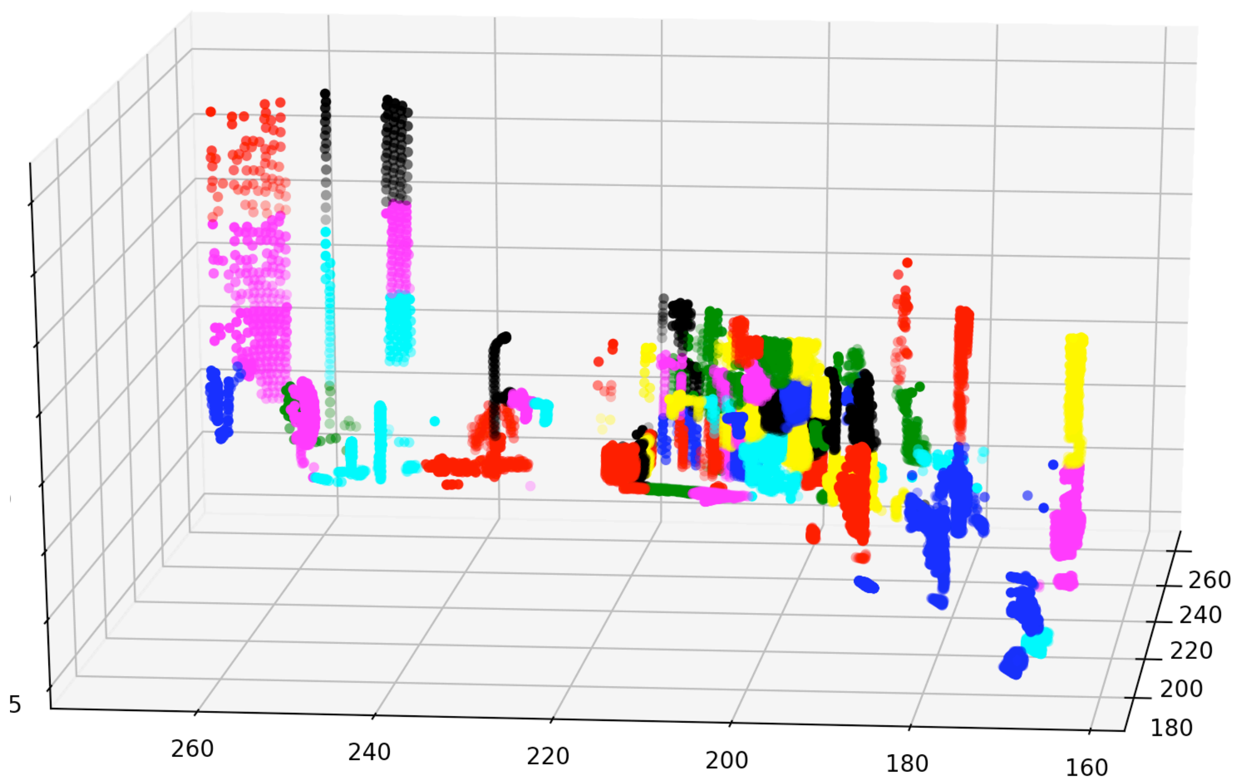
BUILDING FACADE SEGMENTATION RESULT



CLUSTERING

The remaining of the points can now be clustered for classification. The paper suggests an agglomerative clustering with a distance threshold. This has a runtime complexity of $O(n^3)$. For a scene, the number of points to be clustered is around 10^3. This means an agglomerative clustering would result in 10^9 computations. Instead, a k-Means clustering has been used that speeds up performance significantly. Notice that is possible by overestimating the number of clusters followed by a merge into super-voxels to group together similar nearby clusters. For our implementation, clusters sizes of 50-100 generate good results.

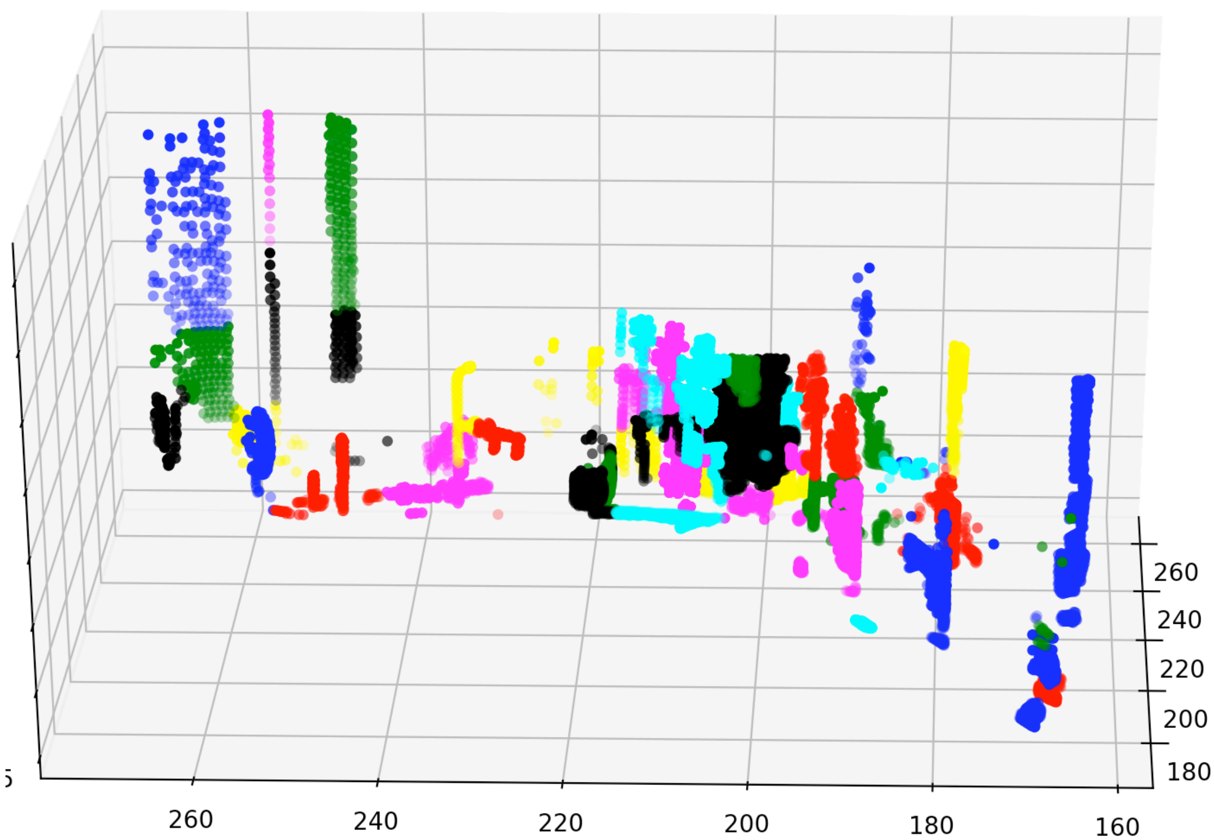
CLUSTERING RESULT



SUPER -VOXEL GENERATION

Once the clusters have been created, we try to merge nearby clusters with similar direction of spread (or surface normals). In this approach as well, Eigen vectors are computed to compute direction of maximum spread. The Eigen vectors of nearby clusters is compared and if they are within a 15 degree threshold, they are merged to create a super-voxel. This reduces the number of clusters by about 33% and generates super-voxels that can be used for classification.

SUPER-VOXEL GENERATION RESULT



CLASSIFICATION

For each of the clusters, the following features are generated:

Geometric features:
area, edgeRatio, maxEdge, Covariance

Orientation and location:
height, normal, density

3D features:
density

The paper suggests a random forest classifier with these extracted features. However, as an improvement, a 3D convolutional net can be directly fed the x, y and z values for learning of 3D spatial features. A ground projection map can be fed into a 2D convolutional net for locational feature extraction and improve classification.

PERFORMANCE REPORT

The code has been tested on a MacBook Pro with the following specifications:

macOS High Sierra
Version 10.13.6

MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports)
Processor 2.3 GHz Intel Core i5
Memory 8 GB 2133 MHz LPDDR3
Startup Disk Macintosh HD
Graphics Intel Iris Plus Graphics 640 1536 MB

The performance report is for execution is as follows for 15 iterations of RANSAC and a cluster size of 100 for kMeans:

```
***********************************************************************
************************* PERFORMANCE REPORT *************************
***********************************************************************

|        MODULE            |      TIME ELAPSED      |    PERCENT OF TOTAL TIME     |
----------------------------------------------------------------------------
  ROAD SEGMENTATION             11 secs                50.000000 %
  BUILDING SEGMENTATION         0 secs                 0.000000 %
  TOTAL CLUSTERING              11 secs                 50.000000 %
 -- KMEANS CLUSTERING           11 secs                 50.000000 %
 -- SUPERVOXEL CREATION         0 secs                 0.000000 %
****************************************************************************
TOTAL TIME ELAPSED: 22
```