

# LiDAR: Analysis and Classification

Steven Spiegel

Wednesday 17<sup>th</sup> April, 2019

## **Abstract**

LiDAR (Light Detection and Ranging) is ubiquitous in its use and applications. Some of its applications is in computer vision and robotics, surveying, emergency management, and object classification. this project aims to introduce the reader into the science of LiDAR, processing tools, and applications. In particular, this paper will look at PDAL (Point Data Abstraction Library) and its use in point cloud processing and application. The point cloud in question is a segment of a North Dakota collect. In conjunction with Python and Jupyter Notebook, this paper seeks to efficiently remove outliers, segment the ground, and classify the remaining points of an airborne LiDAR collection.

# 1 Introduction

LiDAR (Light Detection and Ranging) is used in numerous scientific contexts, particularly in the realm of computer vision and surveying. At its heart, LiDAR measures the distances between transmitted and received signals. There are several main types of LiDAR:

1. Pulse: Emits a pulse and measures the time between hitting a target and the return time.
2. Phase: Measures distances using interferometry.
3. Geiger: Use sensors that are triggered by photons.

The mode of LiDAR used in my project is pulse based [1].

In addition to the different types of LiDAR, there are a number of different collection techniques. The four primary types are terrestrial, airborne, mobile, and Unmanned Aerial Systems (UAS). Terrestrial collects require the use of reflectors or planar regions to register scans together. The main struggle with this type of collect is the amount of work in moving the scanner from different scan positions. However, these scanners often have ranges that are on the kilometer scale, allowing for massive data collects. Mobile scans use the power of Robotic Operating Systems and feature matching to create a point cloud. As the price of small LiDAR scanners and Unmanned Aerial Systems have become increasingly affordable, UAS LiDAR collection is becoming more common. What would usually take days for surveyors to collect with a terrestrial scanner, UAS can collect in hours. A plethora of software suites exist to process the LiDAR data for relative accuracy as well as absolute accuracy.

In this context, we are focused on classifying point clouds once they have been collected and processed. There are a number of techniques to accomplish this. In general, the strategy is as follows: Classify the ground, segment the point cloud, and classify the segments, as demonstrated in the master's thesis of Nina Varney [3]. Other strategies can be taken to classify point clouds, such as PointNet, which utilizes a Convolution Neural Network to segment and classify point clouds [2]. In our case, we directly classify the ground and then use the geometric properties of the remaining points to identify what class they should be in. This is a more simplistic method than shown in [2] and [3], however it will be sufficient for our purposes. We know that in our scene we are only interested in classifying the vegetation and buildings. Hence, we can mainly focus on looking at the geometric properties of the points themselves and then deduce what class they should be in. The general strategy for processing the point cloud is as follows:

1. Remove outliers via a statistical or distance method.
2. Use morphological filtering and return numbers to classify ground returns.
3. Use point covariance to determine normal planes of remaining points.
4. Use Singular Value Decomposition of the Covariance Matrix to determine dimension of the points.

## 5. Classify based on the dimension of points.

In order to accomplish this, we will employ PDAL pipelines which are written as JSON files and used in a Jupyter Notebook [1]. PDAL has a python interface that allows one to create custom python code in addition to PDAL pipelines. These functions are written in C++, but PDAL allows python to be used in addition to the pipelines.

The next section will demonstrate these methods as well as give a brief discussion on the theory of each one. Then, a basic PDAL pipeline will be demonstrated. For the purpose of this exercise, our goal is to classify outliers, ground, buildings, and trees. We will be classifying an airborne LiDAR collect called `CSite1_orig-utm` [1].

## 2 Methods

### 2.1 Outliers

The first thing we need to do is to classify the outliers. Outliers are noise points that aren't attributed to things we are interested in. In airborne LiDAR this could be due to dust, birds, etc. Hence, the first step is to eliminate outliers. There are 2 primary methods: Statistical and Radial.

#### 2.1.1 Statstical

Statistical Outlier Removal requires 2 calculations across the point cloud. On the first pass, for point cloud  $P$  with  $n$  points, and for each point  $p_i$  for  $i \in 1 \dots N$ , the mean distance  $\mu_i$  is computed for  $k$  nearest neighbors. Then, we calculate the global mean of means in the following:

$$\mu = \frac{\sum_{i=1}^N \mu_i}{N} \quad (1)$$

and the standard deviation ( $\sigma$ ) is defined as

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (\mu - \mu_i)^2}{N - 1}} \quad (2)$$

Thus, the threshold ( $t$ ) for outlier classification is as follows:

$$t = \mu + \sigma m \quad (3)$$

where  $m$  is a user defined input. Thus, we have the following criteria for a statistical outlier:

$$outlier_i = \begin{cases} true & \text{if } \mu_i > t \\ false & \text{otherwise} \end{cases} \quad (4)$$

. Hence, any point  $p_i > t$  is considered an outlier and is given the classification of 7 (standard noise class for LiDAR).

### 2.1.2 Radial

The other outlier method is a radial outlier filter. The first calculation involves calculating  $k_i$  nearest neighbors for each  $p_i$ . Then, we select a minimum number of neighboring points  $k_{min}$  and create the following rule.

$$outlier_i = \begin{cases} true & \text{if } k_i < k_{min} \\ false & \text{otherwise} \end{cases} \quad (5)$$

An example pipeline can be found in the appendix ().

The next step is to classify ground returns (classification value of 2). In order to do this, We will employ mathematical morphology with a progressively growing window to find ground returns.

## 2.2 Ground Classification

Morphological filtering dates back to mathematical set theory and digital image processing [4].

## References

- [1] Andrew Bell, Brad Chambers, Howard Butler, and Michael Gerlek. Introduction to lidar - pdal.io. <https://pdal.io/>, 2018.
- [2] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [3] Nina Varney. Lidar analysis for automatic region segmentation and object classification. Master’s thesis, University of Dayton, Dayton, Ohio, 12 2015.
- [4] Keqi Zhang, Shu-Ching Chen, Dean Whitman, Mei-Ling Shyu, Jianhua Yan, and Chengcui Zhang. A progressive morphological filter for removing nonground measurements from airborne lidar data. *IEEE transactions on geoscience and remote sensing*, 41(4):872–882, 2003.