OBJECT DETECTION WITH LIDAR DATASETS

Alvin Liu, Ajay Temal, Kevin Chan

ABSTRACT

Light Detection and Ranging (LiDAR) technology has become essential in applications such as infrastructure and navigation due to its precision in detecting and labeling objects. This paper introduces a machine learning framework for vehicle detection using urban LiDAR datasets in varying, complex environments. We aim to address significant challenges in accuracy and robustness faced by current technologies. Our methodology involves using the Simple Morphological Filter (SMRF) from the PDAL python library to preprocess and filter the LiDAR data in order to segregate ground and non-ground planes. This then allows us to apply our novel Sliding Window algorithm for accurate hole and boundary identification. By successfully bounding target vehicles in the point cloud environment, we are able to train the extracted data through different rigorous validation stages to ensure correct detection. This preliminary method assumes promising accuracy, and ongoing work will focus on refining it to further detect and properly handle erroneous holes.

1. INTRODUCTION

With the rapid rise of automation around the world, the need for efficient and detailed remote sensing technologies has increased. LiDAR technology and hyperspectral imagery has become key components to this movement, by allowing us to work with sparse but informative point cloud data. LiDAR spatial data is collected by emitting laser pulses and measuring the time it takes to reflect back after hitting an object. Through this process, we are able to generate a three-dimensional representation of the scanned surrounding environment. However, LiDAR is repeatedly limited by its ability to identify and categorize objects in the spatial plane making it difficult in cases necessary for pattern recognition and computer vision.

The characterization of object appearances whether in 2D images or 3D spatial datasets has been a well regarded issue and early approaches relied on capturing geometric features such as edges and corners to classify objects which can help boundary analysis techniques. And while we can analyze LiDAR data, it is a time-consuming and labor-intensive procedure that is also prone to human error.
To address these problems, our research presents a framework designed to improve vehicle detection within urban LiDAR datasets in urban areas. Our proposed method consists of ground segmentation, hole detection, and object classification. Ground segmentation separates the ground plane from non-ground points, enabling the identification of candidate holes representing potential vehicles. The hole detection stage employs a sliding window approach and bounding box methodology to determine whether a candidate hole corresponds to a vehicle based on point density and height criteria. Finally, object classification involves training a machine learning model using the extracted vehicle data and validating it against a reference library to accurately detect vehicles in any LiDAR dataset.

In the following sections, we will further discuss the methodologies employed and explore the broader implications of our findings and how we will deal with LiDAR occlusions and data overlaps. We will also outline the development of the data indexing guidance tool and the benchmarking techniques used.

2. BACKGROUND

There has been a large body of work within the object-detection/labeling domain, with relevant research spanning back to the early 2000s. The characterization of the appearance of objects – whether they were over a 2D image format, or within a 3D spatial data set – was dependent on the geometric aspects attributed, such as the edges, corners, and the texture of a given object.

The Viola-Jones object detection framework utilized integral image data structures to effectively capture Harr features, which related to the pixel densities of similarly labeled objects, such as faces or cars (. These were subsequently used to provide a general classification, to which the framework was able to refine through machine learning integration. Delal et al. produced an alternative yet related approach, referred to as the Histogram of Oriented Gradients feature descriptor (HOG), which computed gradient information into histograms in various localized regions of an image to produce a simplified, compact classification of a particular object (Delal, 2005). These descriptors were then utilized as classifiers and were integrated in the testing phase into a sliding window algorithm to detect distinct human "patches" over a variety of images.

The Hierarchical Mixtures of Experts (HME) is a framework that created a unified representation of objects based on a model that was built off hierarchical data (Bastian, 2005). A fundamental aspect of this approach was Generalized Second Moments (GSM) captured second-order data, such as the distribution of pixel density rather than just pixel densities themselves – which within this context could be considered first-order data. These were used to identify images of cars, and the HME integration proved to be substantially more effective at accurate vehicle detection when integrated into a machine learning algorithm when compared to other approaches of its time.

Just to reiterate, the research discussed up to this point has been integrated within their respective studies only on the two-dimensional front. Yet these foundational pieces of research and related studies find their relevance in general pattern detection and object classification. Especially within the work discussed in this paper, the sliding window algorithm is expected to be the primary algorithmic step in object detection, and an HME framework is expected to be integrated into our machine learning integration.

On the GIS spatial analysis front, advancements in boundary analysis in spatial data sets is of particular importance to this research, due to the nature of LiDAR point clouds. Jacquez et al. provide the state of boundary analysis (Jacquez, 2000). This includes various boundary detection techniques, such as edge detection algorithms and boundary segmentation/ extraction. Some of these algorithms utilize moving split windows, triangulation delineation, spatial constrained clustering, and fuzzy set modeling. These algorithms were considered as candidates for the hole detection algorithm to be implemented in our object-detection/labeling process due to the particular importance of identifying boundary points in vehicle detection.

Niemeyer et published an approach similar to what is described in this paper, presented an urban object classification using the context of an urban scene to learn the relationship between objects (Neimeyer, 2005). Niemeyer et al. utilized waveform data, which pertains to the intensity return values and several more characteristics. Within this particular study, objects were classified using both waveform data as well as the background context of an object to accurately classify it.

What isn't discussed much in Niemeyer's approach is how to make such a model robust and applicable to various urban environments. Additionally, the deployment of such an approach is not presented as a practical solution and can face some challenges when attempting to analyze different forms of LiDAR data and/or if it were to be implemented into a wider system for specialized tasks.

This paper outlines our general framework for object detection and labeling in the context of vehicles. Our research wishes to present object detection and labeling with a similar efficacy as Niemeyer et al., however on a wider, more robust scale to deploy a library that can be used across all types of LiDAR data, and within any urban context. Our framework consists of three main algorithmic tasks: ground segmentation, hole detection, and object classification. We provide an in-depth description of our methodology in section 3. In the following section, we will present our approach to the preliminary object detection and labeling algorithm without a machine learning implementation. More on that will be discussed in section 4.

3. SCOPE AND METHODOLOGY

Our work in this paper aims to create a vehicle detection and labeling system on LiDAR data. It can be extremely difficult to pinpoint vehicles in LiDAR datasets of urban areas, as there are a lot of factors that can make LiDAR data noisy. LiDAR data collected in urban areas will often feature a lot of elevation variation due to the presence of buildings and trees. In addition, data collected may be incomplete due to moving objects, or light that is never reflected back. These factors can make detecting specific objects, such as vehicles, very difficult. A smarter and less time-consuming method would be to flip the LiDAR dataset upside down, which now removes the noise and height variance from the data, and leaves behind a flat plane. Since LiDAR data is collected through light reflectance, only true ground will contain LiDAR data points, and all other objects that have elevation will instead leave behind a concave hull. Splicing out the ground data from the rest of the LiDAR point cloud data will leave behind object-shaped holes in the dataset in place of these concave hulls, which we can detect and determine the shape of. Once these holes have been detected, then we will be able to identify and label vehicles from these hulls, and then use them in order to create a machine learning reference library.

3.1 OVERVIEW

Our approach to vehicle detection on LiDAR point cloud data of urban areas is based on the detection and identification of holes in the ground dataset. There are three major steps and a fourth optional one in our proposed methodology, which have been detailed below:

1. Data Filtering and Preprocessing

   The first step in our proposed methodology is to clean up the LiDAR data so that we will be able to implement our vehicle detection algorithm on the dataset. Since we will be performing vehicle identification by finding holes within the ground from point cloud data that has been flipped upside down, the ground data will need to be isolated and returned on its own. This will be done using the Simple Morphological Filter (SMRF) implemented using the PDAL python library.

2. Vehicle Identification

   After the data has been preprocessed, we will be able to identify candidate holes within the segmented ground returns. Our approach towards hole identification will be using the sliding window method, which will iterate over the dataset with a moving window. Once the hole boundaries have been defined, we propose to use the bounding box methodology, which will test for point density and height in order to determine whether or not a candidate hole is a vehicle.

3. Data Retrieval

   Once the vehicles have been identified, the next step is to retrieve the vehicle boundaries from the dataset. Our implementation of the sliding window algorithm should leave only the points around hole boundaries behind, so retrieving the vehicle boundaries from the data should be relatively simple.

4. Training and Validation

   The final step in our proposed methodology is to train a machine learning model using the extracted data and validate against a reference library. The final goal is to be able to detect all vehicles within any LiDAR dataset, but this final step is not within the current scope of the project.

We were able to lay out the full plan for steps one and two, but due to time constraints and general inexperience, we were unable to implement the aforementioned steps in action during the timeframe of this project.

3.2 PREPROCESSING

Before we can detect vehicles in LiDAR point clouds, the data must be filtered and segmented in order to detect holes in the ground plane. There are a few different methods in order to separate the ground plane from the rest of the LiDAR dataset. One such method is the Simple Morphological Filter (SMRF) algorithm, and this was our method of choice. SMRF is a straightforward technique in segmenting point cloud data into ground and non-ground points. The algorithm is based on the principle that ground points form continuous surfaces with relatively uniform elevation values, whereas non-ground points tend to have greater variability in elevation values. SMRF consists of three steps - creating a minimum elevation surface map from the point cloud data, segmenting the surface map into ground and non-ground grids, and then finally segmenting the actual point cloud data. (PDAL Contributors, 2022)

In order to abstract the process, we decided to use the PDAL python library. Point Data Abstraction Library, or PDAL for short, is an open source library for processing point cloud data, and with python support point cloud data can be processed with PDAL into Numpy arrays. SMRF is a built-in filter in PDAL, and using filters.smrf, the SMRF algorithm can be used with PDAL in order to classify ground and non-ground returns without needing to manually complete the SMRF algorithm and create elevation maps. The first step in segmenting the ground data is to create a PDAL pipeline, which is a sequence of operations to be applied to the point cloud data enclosed within a JSON object. Within the pipeline, we will invoke a call to filters.smrf within the JSON, as well as other filter contents, which are detailed in Bradley Chambers' PDAL ground filter tutorial. (Chambers, 2017) Once the ground and non-ground data have been classified with SMRF, the data can easily be segmented using a range filter, which filters points based on classification values. By default, point cloud datapoints have a classification value equal to zero, which have been detailed in the PDAL pipeline. Once SMRF is run over the data, points which have been determined to be ground points will have a classification value of two. We can then apply a range filter using PDAL in order to extract only ground data by filtering for points with a classification value of 2. Figure 1 shows an example image of a manually segmented ground plane, and we expect that the ground plane segmented using SMRF will look similar.

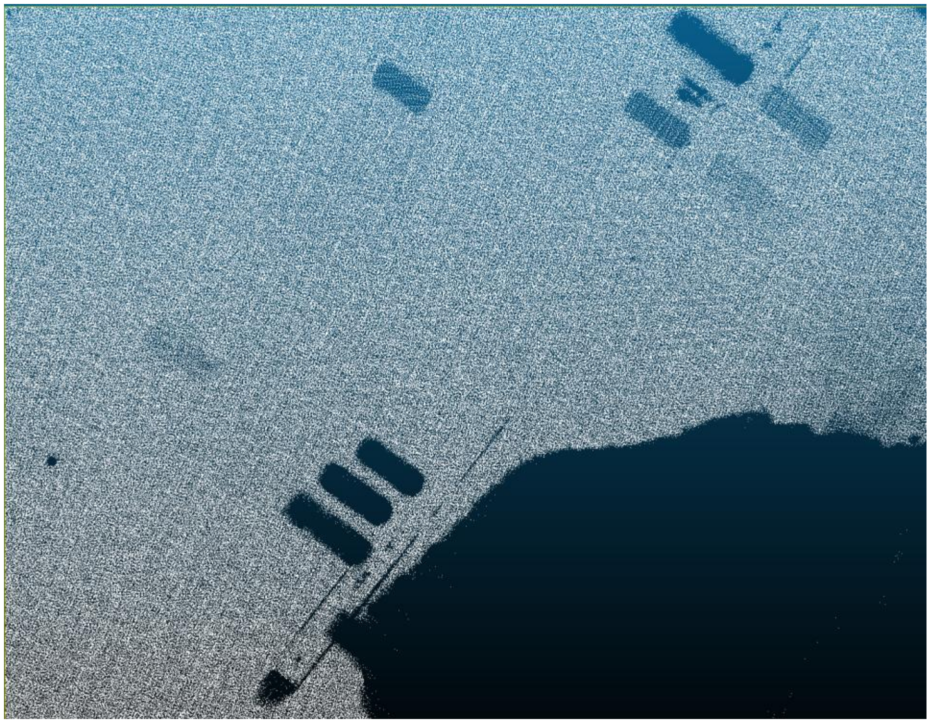

Figure 1: Manually segmented ground plane from sample LiDAR point cloud data. Note the holes in the data representing the silhouette of vehicles.

3.3 SLIDING WINDOW

With an exported ground plane, we can now apply a hole detection algorithm. To reiterate, the goal of this algorithm is to detect a vehicle based on the hole it creates from standard LiDAR scans. For its robust and scalable nature, the sliding window algorithm was chosen to scan over the ground plane. A visual representation of this algorithm's setup is provided by Figure 2. The window's size is yet to have an exact value; however, it's been established to be less than the expected area that the standard vehicle covers. To balance our accuracy and efficiency, the window will move half of its length for each iteration. For each iteration, the window will make a decision based on the point density of the area it is currently over. This parameter gives reason to the requirement to have the window size be less than the standard vehicle. If it were larger, both high-density and low-density areas could be in the window at a given time. This would make it harder to have a dependable density value to discriminate on.

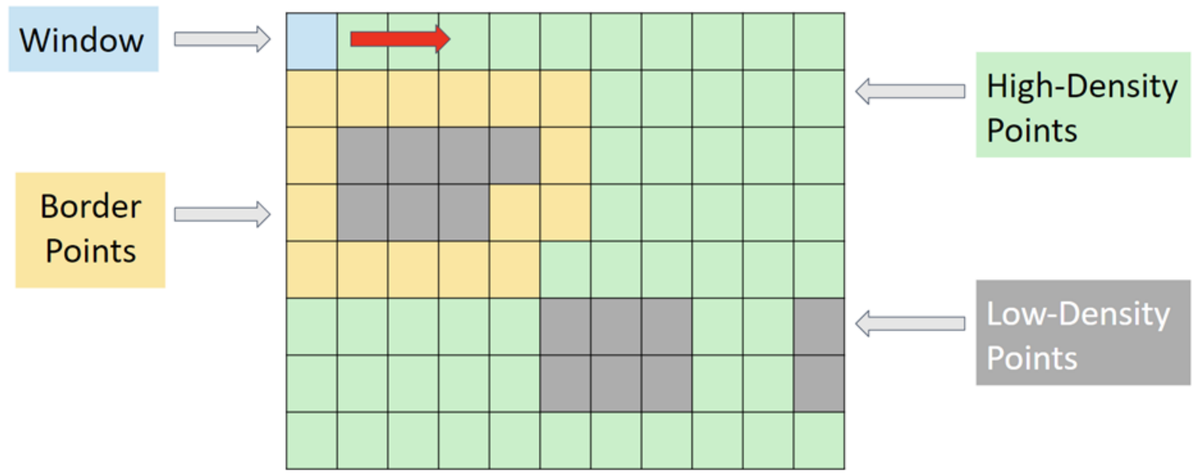

Figure 2: Visual representation of ground plane and moving window

As mentioned before, boundary analysis is of particular importance to this algorithm, and there are two reasons for that, as there are two scenarios in which the window encounters a low-density area. The window could be extended over the boundary of the ground plane, which is areas not to be considered. Or, the window could be over a hole within the bounds of the ground plane, which is what it is exactly looking for. The window should only produce two different actions based on which type of low-density area it encounters. To distinguish between these two types of low-density areas, the window's behavior will be altered. The altered behavior later will also help the algorithm define the holes, if it is established by the behavior that the window is currently over a hole. At low-density points, the window will expand outwards in all directions except for the right, demonstrated by Figure 3. Beyond a certain threshold, if the window has not reached a high-density area, we know that the window is out of the ground plane's bounds. This threshold will be dependent on the anticipated area the car could take in any possible orientation.

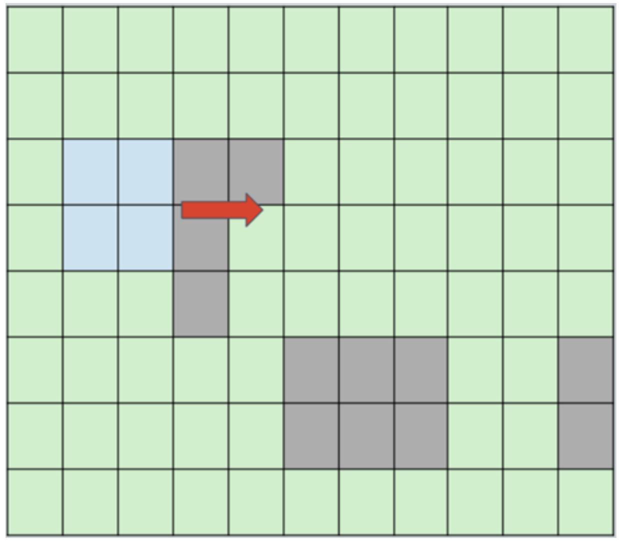

Figure 3: Window expanding in hole

Now, if it is a low-density area that indicates that the window has reached the left boundary of the ground plane, the window must reset to the next row to iterate over. If the area is determined to be a hold, the expanding window will be used to collect the border points of the hole. The border points are crucial for establishing the bounds of the hole. To get to the border points, the window will expand until it reaches a density threshold. From there, the window can move in several directions until it encounters more high-density areas. These border points will then be segregated and marked within the point cloud. The window will continue to slide over until it has covered all points within the bounds of the point cloud. It is expected that this algorithm returns the locations of each valid hole in the point cloud.

3.4 BOUNDING BOX

Having received the segregated holes, the object identification algorithm will need to cross reference the original point cloud's points at the location that the holes were segregated from the ground plane in the previous step. This algorithm will then create a bounding box in the area in the original point cloud where the hole has been detected. This bounding box will be used to test whether the hole returned is the location of a vehicle. This is necessary because although the hole detection algorithm tests with parameters based upon common vehicle characteristics, there can be exceptions, such as in Figure 4.

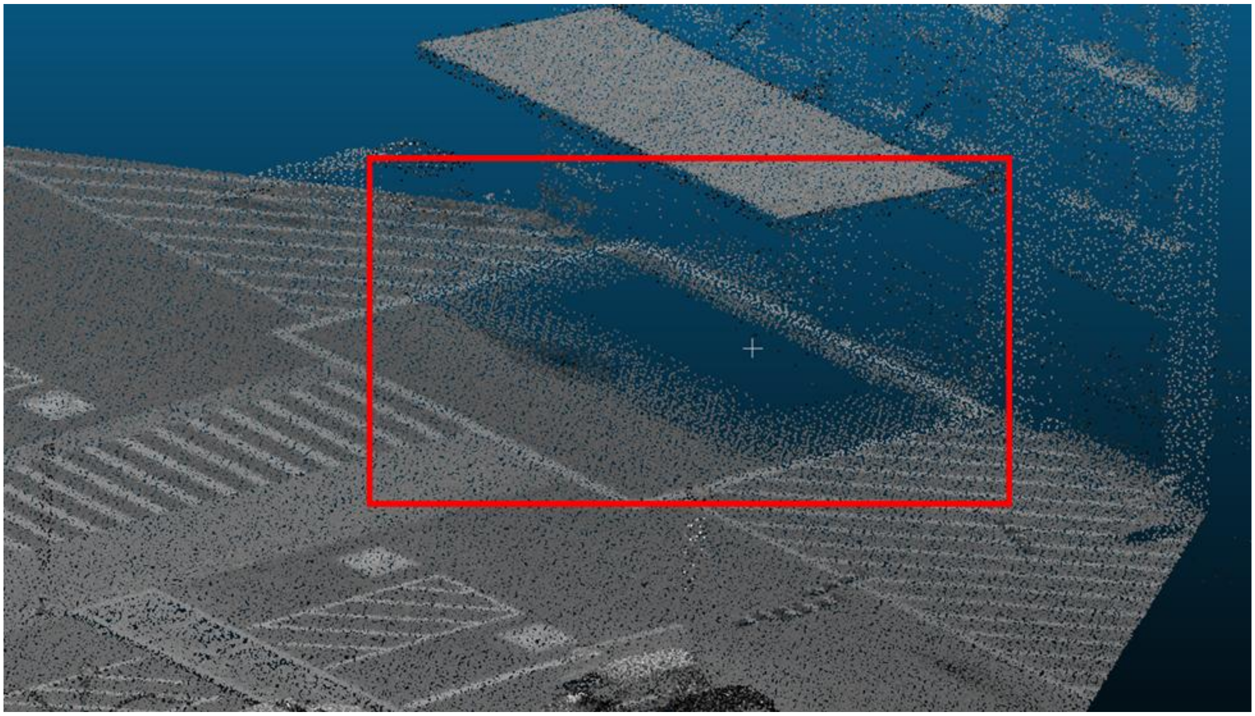

Figure 4: Hole caused by building overhang

The bounding box will test for two parameters: height and point density. While not all vehicles are the same height, there is a reasonable range to expect a vehicle to be. Due to the variety in vehicle height and the possibility for other objects to be hanging over a vehicle, such as a tree branch, concentrated point density will have to be considered as well. Within the box, if most of the point density is centered at the height generally expected of a vehicle, we can expect it to be a vehicle.

Of course, the bounding box algorithm, without any deep learning integration, appears to be prone to some degree of error, due to the variety in vehicle design and a vehicle's surrounding context. Further integration will be discussed in the next section.

This object identification algorithm will be the final return of the process, returning a bounding box from the original point cloud that has satisfied its parameters. Figure 5 displays what a return could look like.

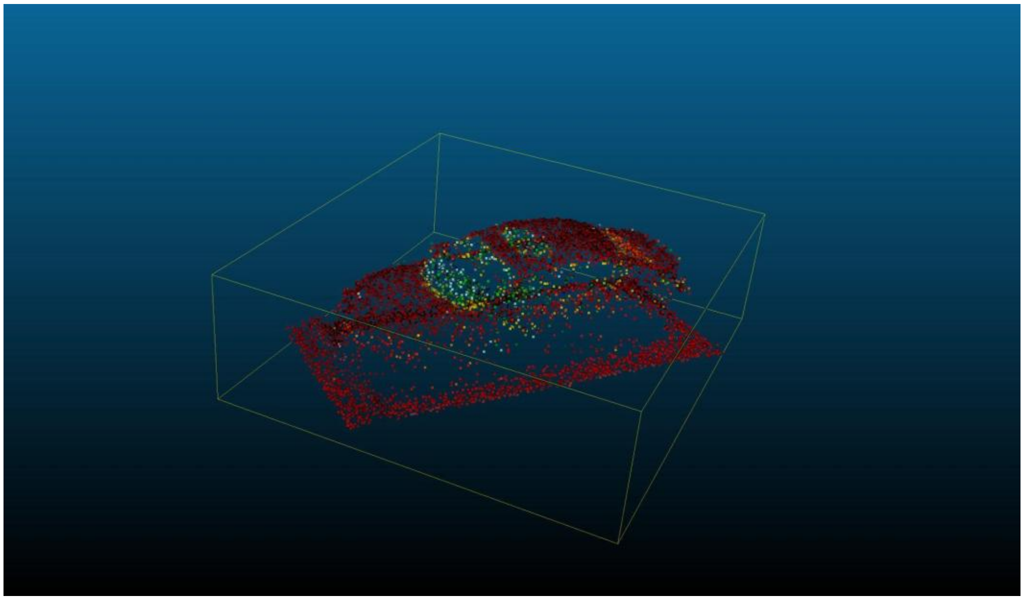

Figure 5: Example bounding box with vehicle

4. NEXT STEPS

The groundwork for executing our goal of vehicle detection from LiDAR datasets has been laid out, and so our next step is to implement the methodology in action. The first step will be to filter the data and segment the ground from the sample LiDAR point cloud dataset using our proposed methodology, in which we will implement the SMRF algorithm via the PDAL python library. Once we have the ground data, we will be able to implement the sliding window algorithm and bounding box method in order to detect vehicles within the dataset. Some fine tuning will be needed with regards to sliding window parameters such as step size, window size, and density threshold. Once hole boundaries have been found, we will need to find a candidate bounding box library in order to determine if a hole is a vehicle or not. The sliding window algorithm will ideally be able to remove all data points except for the hole boundaries, which will make data extraction a relatively simple task. In the future we also plan to create a machine learning model and reference library using the extracted vehicle data as referenced in section 3.1. We will need to search for a candidate training library that can be applied to the entire non-segmented full dataset, which may allow us to find cars that may not have been found through our current proposed methodology, which may leave out cars that have changed positions between snapshots, leading to incomplete data.

5. CONCLUSION

This study has demonstrated our scalable framework for vehicle detection in LiDAR point cloud datasets, addressing the underlying challenges we face with ambiguous data. Utilizing our Sliding Window algorithm for hole and boundary detection, we were able to address and mitigate issues such as holes of variable densities in the preprocessed data after running it through the PDAL filter. Although further refinement is needed, the proposed framework demonstrates promising potential for enhancing object detection and labeling as we start building our machine learning pipeline for training and validating objects of diverse datasets. For the future, our focus will be on improving detection capabilities to ensure broader applicability to other objects by optimizing the sliding window technique, exploring alternative boundary analysis techniques, and applying different deep learning models to compare against.

WORKS CITED

Bastian, Seemann, Schiele. “Learning Appearance-Based Models: Hierarchical Mixtures of Experts Approach Based on Generalized Second Moments.” European Conference on Computer Vision, 2004. https://www2.eecs.berkeley.edu/Pubs/TechRpts/1996/5360.html

Chambers, Bradley. “Ground Filter Tutorial.” *pdal.io Documentation*, 17 April 2017, https://pdal.io/en/2.6.3/tutorial/ground-filters.html

Jacquez, G., Maruca, S. & Fortin, MJ. “From fields to objects: A review of geographic boundary analysis” , 2000. https://doi.org/10.1007/PL00011456

Dalal, Triggs. “Histograms of Oriented Gradients for Human Detection.” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. https://ieeexplore.ieee.org/document/1467360

Niemeyer, J., Wegner, J.D., Mallet, C., Rottensteiner, F., Soergel, U. “Conditional Random Fields for Urban Scene Classification with Full Waveform LiDAR Data”, 2011 https://doi.org/10.1007/978-3-642-24393-6_20

PDAL Contributors, 2022. PDAL Point Data Abstraction Library. https://doi.org/10.5281/zenodo.2616780

Viola, Paul, Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features.” February 2001 https://www.researchgate.net/publication/3940582_Rapid_Object_Detection_using_a_Boosted_Cascade_of_Simple_Features