Pole-Shaped Object Extraction from LiDAR Point Clouds of Roads

Danesh Shokri[1], Shirin Malihi[2], Saeid Homayouni[3], Christian Larouche[4], Heidar Rastiveis[5], Ioannis Brilakis[6]

[1]Département des Sciences Géomatiques,Université Laval, Canada
[2]Civil Engineering Department, University of Cambridge, United Kingdom
[3]Centre Eau Terre Environnement, Institut National de la Recherche Scientifique, Canada
[4]Département des Sciences Géomatiques, Université Laval, Canada
[5]Civil and Construction Engineering, Oregon State University, USA
[6]Civil Engineering Department, University of Cambridge, United Kingdom

[1]danesh.shokri.1@ulaval.ca, [2]sm2852@cam.ac.uk, [3]saeid.homayouni@inrs.ca, [4]christian.larouche@scg.ulaval.ca, [5]heidar.rastiveis@oregonstate.edu, [6]ib340@cam.ac.uk,

Abstract -

Mobile Laser Scanner (MLS) technology enables the acquisition of high-precision three-dimensional lidar point cloud data of roadside infrastructure elements, including traffic signs and street poles. This research presents an innovative and computationally efficient algorithm for the extraction of pole-shaped objects from MLS point clouds. The methodology addresses the computational challenges associated with large-scale point cloud processing through a trajectory-based segmentation and ground point filtering. The algorithm implements geometric descriptors, specifically linearity and verticality to identify potential pole structures. Subsequently, the statistical outlier removal (SOR) filter is applied to refine the candidate pole detection results. The final classification of true pole objects is achieved through the application of ground clearance criteria. The algorithm's effectiveness was validated through empirical testing across three distinct urban and suburban environments, utilizing data acquired from two different MLS systems. The experimental results demonstrated successful performance metrics, achieving mean precision and recall rates of 97.21.

Keywords -

Road, Linearity, Verticality, Trajectory, LiDAR, Poles.

1 Introduction

Poles along roads are really important for new technologies like self-driving cars and creating 3D maps [1]. By knowing exactly where these poles are located, we can better understand road conditions and improve how we design and manage transportation infrastructure [2]. While traditional inspection methods involving manual ground-based assessment are resource-intensive and laborious, there is a critical need for expedited, fully automated inspection protocols for roadside infrastructure evaluation [3].

MLS systems have emerged as powerful tools in this field, offering significant advantages over other remote sensing technologies [2]. Unlike optical imaging, which struggles with illumination variations, laser scanning systems can measure distance information directly without being adversely affected by ambient lighting or shadows [4] , [5]. These systems generate three-dimensional spatial information of object surfaces, eliminating the time-intensive computational processes typically required for deriving 3D coordinates from imagery [6].

Previous research has explored various techniques for extracting pole-shaped objects from point clouds, including voxel-based procedures and adaptive radius cylinder models [7]. However, existing approaches often face limitations such as dependence on complex voxel transformations, suboptimal thresholding parameters, and requirements for extensive training datasets and computational resources.

In response to these challenges, the research proposes a novel and computationally efficient algorithm for pole-shaped object extraction from mobile laser scanning point clouds. The methodology introduces geometric descriptors for pole detection while maintaining computational efficiency through systematic data segmentation. By addressing the limitations of existing approaches, this research aims to advance automated infrastructure monitoring and environmental mapping technologies.

2 Proposed Method

2.1 Pre-processing

This subsection addresses the critical challenge of managing the vast volume of lidar point cloud data. It emphasizes the importance of effectively removing both noise and ground points, which is essential for reducing the overall data size and ensuring that only relevant information is retained. This process plays a key role in minimizing unnecessary data, thereby enhancing the efficiency of subsequent data processing and analysis.

2.1.1 Sectioning

MLS systems leverages trajectory data to efficiently segment point cloud information by recording the vehicle's precise location throughout the scanning process. This trajectory data, which is inherently aligned with the point cloud coordinate system, allows for seamless segmentation of the scanned environment into uniform sections without requiring complex georeferencing. By dividing the trajectory into fixed-length segments—in this case, 90m—the corresponding point cloud data is simultaneously partitioned into equivalent sections, creating a systematic approach to processing large spatial datasets [2]. The primary advantage of this method lies in its ability to transform continuous, complex point cloud information into manageable, discrete units, significantly reducing computational complexity and enabling more focused spatial analysis. Since the trajectory data is substantially smaller in volume compared to the lidar point cloud data, this segmentation approach not only simplifies data processing but also improves overall implementation efficiency, making it an elegant solution for handling extensive MLS datasets. Figure 1-a illustrates the trajectory data overlaid on the collected LiDAR point clouds. It is important to note that an elevation shift is required due to the absence of elevation data in the trajectory. As shown in Figure 1-b, the sectioning of the collected data has been appropriately performed, with each section represented in a distinct color.

subcaption

xcolor

2.1.2 Noisy Points Removal

Noisy points (Figure 2) in MLS point clouds can arise from various sources, such as sensor imperfections, environmental noise, and surface reflectivity variations. These points typically exhibit abnormal elevations or low point density, which makes them distinguishable from the rest of the point cloud. The SOR method is a robust technique used to detect and eliminate these outliers by analyzing the statistical distribution of neighboring points [8].

The SOR algorithm operates by calculating the mean distance from each point to its neighboring points within a specified radius. In this study, the number of nearest neighbors (N) is set to 10. For each point in the point cloud, the mean distance to its 10 nearest neighbors is computed (Obtained based on trial and error). These mean distances are then used to define a threshold, which is typically based on the global mean and standard deviation of the mean distances across the entire dataset. Points that have a mean distance significantly larger than the average (i.e., those that fall outside a certain multiple of the standard deviation) are considered statistical outliers. These outliers are likely to be noisy or erroneous, as they do not align with the expected spatial distribution of neighboring points. The general formula for the SOR algorithm can be expressed as [8]:

$$D_i = \frac{1}{N} \sum_{j=1}^{N} |P_i - P_j| \tag{1}$$

where D_i is the mean distance from point P_i to its nearest neighbors P_j, and N is the number of neighbors (in this case, 10). For each point P_i, the algorithm calculates the mean distance to its neighbors and compares it to the overall distribution of mean distances in the dataset. Points with a mean distance greater than a threshold defined as $\mu + k\sigma$ (where μ is the mean and σ is the standard deviation of all mean distances, and k is a defined constant) are flagged as outliers.

xcolor

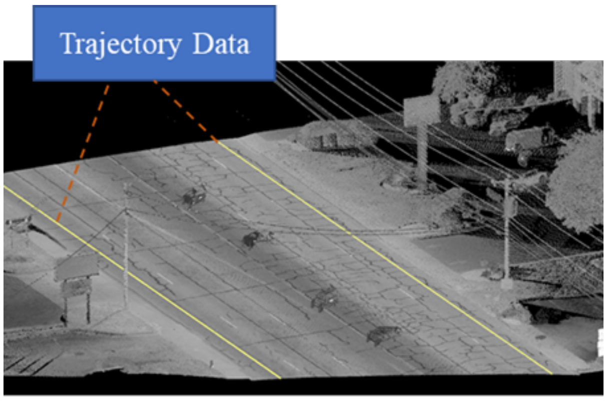
2.1.3 Ground Points Removal

The Cloth Simulation Filter (CSF) method [9] offers an innovative approach to ground point removal in MLS point clouds, addressing the critical challenge of data volume reduction for efficient processing and analysis. By simulating a cloth-like surface that drapes over the point cloud, the algorithm mimics gravitational effects, effectively classifying points as ground or non-ground based on their position relative to this simulated surface. This method demonstrates remarkable versatility, capable of handling diverse terrains including both steep and flat areas, making it particularly valuable for large-scale point cloud datasets.

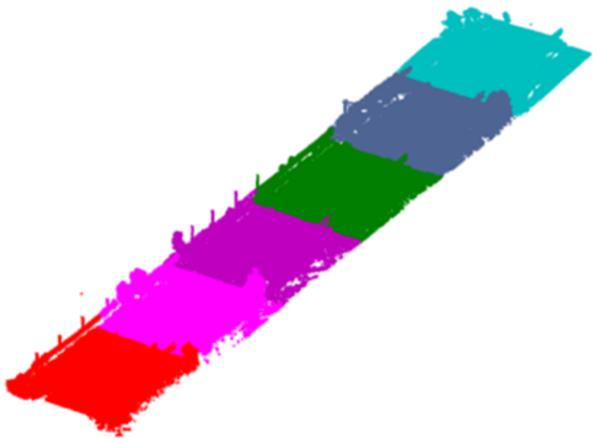
The CSF algorithm's effectiveness hinges on two key parameters: grid size and classification threshold. The grid size, which determines the resolution of the cloth mesh, plays a crucial role in capturing terrain features accurately—in this study, a 0.31m grid size was selected to optimize detail and efficiency. By carefully tuning these parameters, the method can dramatically reduce point cloud volume, as evidenced by its ability to remove approximately 4.5 million ground points from a 5-million-point dataset in just 3 seconds. This computational efficiency, combined with the algorithm's ability to model complex terrain variations, makes the CSF method a powerful tool for preprocessing lidar point cloud data across various environmental contexts.

2.2 Detecting Candidate Poles

This step aims to identify pole-shaped objects among the remaining point clouds, which include various objects



(a) Displaying the trajectory data on the LiDAR point clouds (an elevation shift is needed).



(b) Created sections which partition the point cloud for efficient processing.

Figure 1. (a) Trajectory data overlaid on the LiDAR point clouds, requiring an elevation shift for alignment, and (b) resulting sections dividing the data into manageable units.

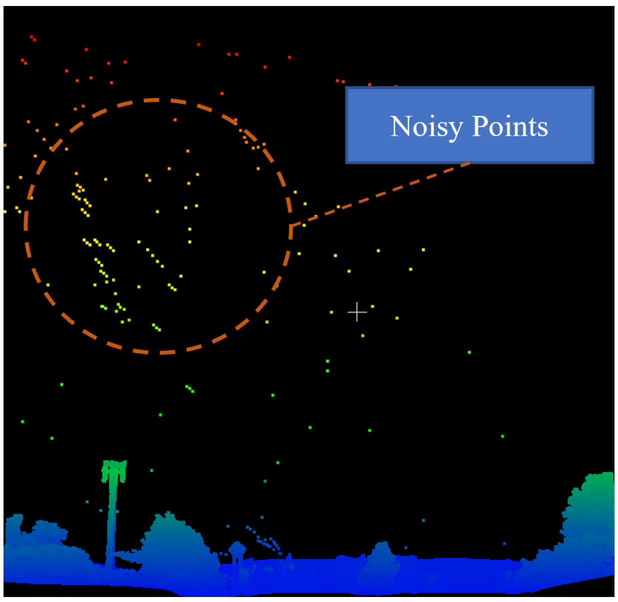


Figure 2. Displaying the noisy points.

that lack meaningful information in isolation. By analyzing the surrounding neighborhood points, we aim to assign meaningful descriptors to each point. For instance, we know that power lines typically exhibit thin, lengthy linear structures, while objects such as poles tend to have a long, vertical shape. To distinguish between these types of objects, rule-based descriptors of linearity and verticality are employed. These two geometric features—verticality and linearity—are particularly effective for characterizing poles. The descriptors are quantified using the eigenvalues obtained from the Principal Component Analysis (PCA) procedure, which captures the directional properties of the point cloud and provides a means to differentiate between vertical and linear structures.

These descriptors are computed for each point in the filtered point cloud dataset after ground point removal. Let $P(i)$ represent the set of points, where i is the index of each point. The descriptors are computed based on eigenvalues derived from PCA of the covariance matrix.

2.2.1 Verticality and Linearity Descriptors

The verticality descriptor $V(i)$ is defined as [2]:

$$V(i) = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \tag{2}$$

where λ_1, λ_2, λ_3 are the eigenvalues of the covariance matrix, and λ_1 is associated with the dominant vertical direction of the point cloud.

The linearity descriptor $L(i)$ is computed as:

$$L(i) = \frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

where λ_2 corresponds to the direction of the linearity of the point cloud.

2.2.2 Neighborhood Selection and Thresholding

For each point index i, the neighborhood points are selected based on a spherical neighborhood with radius $r = 0.9$ meters. This can be expressed as:

$$N(i) = \{P(j) \mid \|P(i) - P(j)\| \leq r\} \tag{4}$$

where $N(i)$ is the set of neighboring points for point $P(i)$, and r is the radius for point selection.

2.3 DBSCAN-based Point Cloud Clustering

To detect pole-shaped objects in a point cloud using an unsupervised approach, we employ the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, which clusters points based on their geometric characteristics [10]. The input to DBSCAN consists of feature vectors for each point, which include the 3D coordinates as well as additional descriptors, such as linearity and verticality. Each point $P(i)$ is represented by a feature vector:

$$\mathbf{x}(i) = \{x_i, y_i, z_i, \text{linearity}_i, \text{verticality}_i\} \tag{5}$$

where x_i, y_i, z_i denote the 3D coordinates of point $P(i)$, and linearity_i and verticality_i are the calculated descriptors based on the local neighborhood and PCA. These descriptors serve as indicators of whether a point is part of a pole-like object or not.

The DBSCAN algorithm requires two parameters: the maximum distance ϵ and the minimum number of points MinPts. The distance threshold ϵ determines the maximum allowed distance between two points for them to be considered as neighbors, and MinPts defines the minimum number of points required in a neighborhood for a point to be classified as a core point. The neighborhood of a point $P(i)$ is defined as the set of points $N_\epsilon(i)$ within a distance of ϵ:

$$N_\epsilon(i) = \{P(j) \mid \|\mathbf{x}(i) - \mathbf{x}(j)\| \leq \epsilon\} \tag{6}$$

where $\|\mathbf{x}(i) - \mathbf{x}(j)\|$ is the Euclidean distance between points $P(i)$ and $P(j)$. If the number of points within this neighborhood is greater than or equal to MinPts, then $P(i)$ is considered a core point, which is a point around which a cluster can form. This is mathematically represented as:

$$|N_\epsilon(i)| \geq \text{MinPts} \tag{7}$$

Points that lie within the neighborhood of a core point but do not themselves have enough points to form a cluster are classified as border points. Border points are included in the same cluster as the core points they are associated with. Any point that is not a core point or a border point is classified as noise.

DBSCAN assigns each core point and its corresponding neighborhood to a cluster, and the final output is a set of clusters $C_1, C_2, \ldots, C_k$, where each cluster contains densely connected points. Points that are classified as noise are not included in any cluster. The overall output of the DBSCAN algorithm is given by:

$$\text{DBSCAN}(\epsilon, \text{MinPts}) = \{C_1, C_2, \ldots, C_k\} \tag{8}$$

where $C_1, C_2, \ldots, C_k$ are the detected clusters corresponding to potential pole-shaped objects. This clustering approach enables the identification of pole-like structures based on their geometric properties without the need for labeled data.

3 Experiment and Results

3.1 Study Area and Results

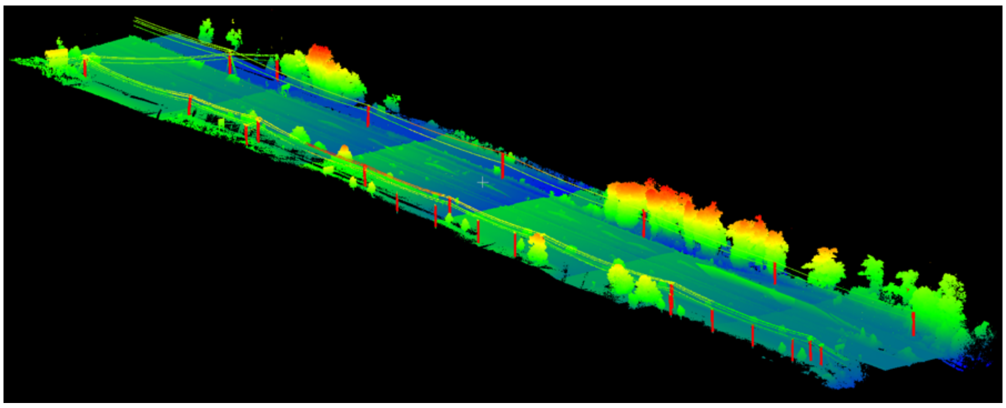
The evaluation of the proposed algorithm is based on datasets from diverse environments, each presenting unique challenges for accurate analysis. The urban region in the USA, with its high object density and reflective surfaces, complicates the distinction of pole-shaped structures due to overlapping vertical objects, occlusions, and varying point densities. The suburban USA region, while less dense, still presents challenges with trees, power lines, and scattered buildings, requiring the algorithm to manage a large dataset with 45 million points and efficiently classify pole-shaped objects. The suburban region in China adds complexity due to geographical differences, sensor calibration variations, and environmental factors like vegetation and local infrastructure. Across all datasets, challenges include point density variations, occlusions, environmental factors, and geographical variability, which demand a robust, adaptive algorithm.

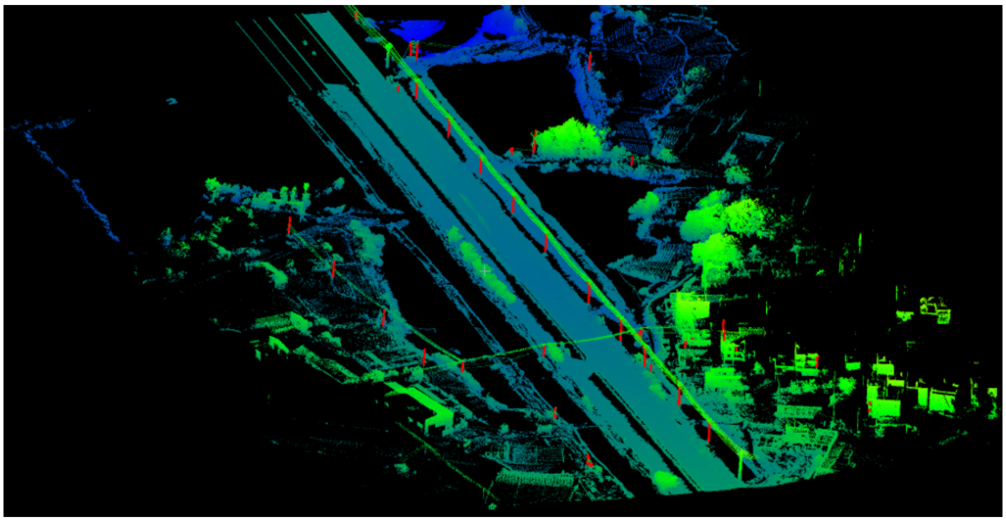
The algorithm demonstrated strong performance (Figure 3) across the test regions, particularly in the suburban USA area where it successfully detected 33,490 pole points with only 796 points undetected and 953 false positives, achieving approximately 97 percent recall accuracy (Table 1). In urban environments, additional challenges arose from phone lines attached to poles and traffic signs, which occasionally led to misclassification. While the algorithm showed reduced accuracy in detecting transmission line pylons and poles with additional equipment such as transformers, its adaptability to various environmental conditions proved effective. The presence of dense trees, cables, and varied building types added complexity to the detection process, yet the algorithm maintained robust performance across different regions, efficiently processing large lidar datasets and adapting to diverse urban and suburban landscapes.

4 Discussion

4.1 Computation System

The proposed method presents significant advantages over previous approaches, particularly in terms of system implementation, runtime efficiency, input data requirements, and overall efficiency. Unlike many prior studies that rely on expensive cloud computing systems, the method was executed on an affordable laptop featuring an Intel Core i5 processor, 12 GB of RAM, and an Nvidia GeForce graphics card. This economical setup still


(a) Suburban USA


(b) Suburban China

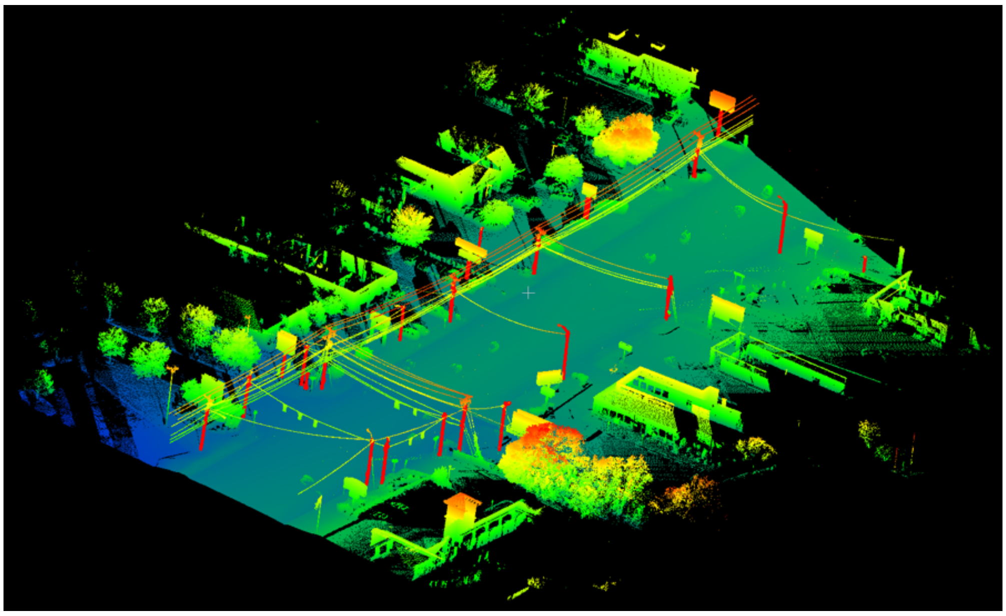

(c) Urban USA

Figure 3. Visualising the output of the algorithm (the extracted poles are displayed in red color): (a) Suburban USA, (b) Suburban China, (c) Urban USA.

Table 1. Performance metrics of the proposed algorithm across three datasets, with average values.

Dataset	Length (m)	Points (million)	Precision (%)	Recall (%)	F1 Score (%)	Time (s)
Suburban USA	550	45	97.68	97.23	97.45	104
Suburban China	500	19	97.00	95.84	96.42	93
Urban USA	200	15	96.96	94.97	95.95	115
Average	–	–	97.21	96.01	96.61	–

achieved satisfactory performance, making it accessible for practitioners without high-end infrastructure, and particularly suitable for resource-constrained environments like field-based applications or small-scale projects. Additionally, the runtime efficiency of the method is a key strength, processing data in approximately 312 seconds, which is significantly faster than other methods—up to 20 times faster in some cases.

4.2 Datasets

In terms of input data, the method has the advantage of relying solely on 3D lidar point clouds and trajectory data, simplifying the data acquisition process compared to other methods that require additional data sources such as scanlines or labeled training datasets. This makes the algorithm more flexible and easier to implement in situations where such data is difficult or costly to obtain. The method's versatility is further demonstrated by its ability to handle data from multiple MLS platforms and operate effectively in both urban and suburban environments. This adaptability enhances the algorithm's robustness and broadens its applicability across diverse geographic locations and point cloud characteristics. Overall, the proposed algorithm outperforms existing methods by providing high performance on affordable hardware, efficient processing, minimal data requirements, and flexibility across different platforms, making it a practical and valuable tool for pole detection in lidar point clouds.

4.3 Parameter Selection and Optimization

The effectiveness of the proposed algorithm relies on several key parameters, whose values were carefully selected to optimize performance across diverse MLS datasets. While the primary parameters—segmentation length (90 m), SOR nearest neighbors ($N = 10$), and CSF grid size (0.31 m)—were justified in the methodology, additional parameters such as the spherical neighborhood radius ($r = 0.9$ m), DBSCAN clustering parameters ($\epsilon = 0.5$ m, MinPts = 10), and ground clearance threshold (0.5 m) also warrant discussion to elucidate their optimality.

The spherical neighborhood radius of 0.9 meters, used for computing verticality and linearity descriptors, was chosen to encapsulate the local geometry of pole-shaped objects while minimizing interference from adjacent structures. Given that poles in the datasets typically have diameters of 0.2–0.5 meters, a 0.9-meter radius ensures sufficient point inclusion for robust PCA-based eigenvalue computation, capturing vertical extents of 2–3 meters. Empirical trials showed that smaller radii (e.g., 0.5 m) yielded unstable descriptors in sparse regions, whereas larger radii (e.g., 1.5 m) incorporated noise from nearby objects, reducing specificity. This value proved effective across urban and suburban environments, balancing sensitivity and precision.

For DBSCAN clustering, the maximum distance $\epsilon = 0.5$ m and minimum points MinPts = 10 were selected to group points into coherent pole structures. The 0.5-meterϵ aligns with the physical scale of poles and the point density of the MLS data (100–1000 points/m^2), ensuring that points along a single pole are clustered without merging with unrelated objects spaced further apart. A smaller ϵ (e.g., 0.2 m) fragmented poles, while a larger value (e.g., 1.0 m) over-clustered with non-pole features. Similarly,MinPts = 10 reflects the expected point count within a pole's neighborhood, filtering out noise while retaining valid clusters. This combination was iteratively refined to maximize recall and precision, as evidenced by the 97.21% average precision reported in Table 1.

The ground clearance threshold of 0.5 meters, applied as a final classification criterion, distinguishes pole bases from ground-level clutter. This value accommodates terrain variations and sensor height offsets, ensuring that poles are correctly identified above residual ground points post-CSF filtering. Testing revealed that a lower threshold (e.g., 0.2 m) misclassified low vegetation, while a higher threshold (e.g., 1.0 m) missed valid pole bases in uneven areas. The 0.5-meter criterion proved robust across the datasets, enhancing the algorithm's adaptability to diverse landscapes.

These parameter choices were informed by empirical optimization, practical constraints (e.g., computational efficiency on standard hardware), and the physical characteristics of pole-shaped objects in MLS point clouds. While the selected values yielded high performance (average F1 score of 96.61%), future work could explore adaptive parameter tuning based on local point density or environmental context to further enhance generalizability. Nonetheless, the current settings provide a practical and effective baseline for pole detection, as demonstrated by the experimental results.

4.4 Comparison with Baseline Methods

The proposed approach, leveraging trajectory-based segmentation and geometric descriptors, is designed to offer improved efficiency over existing techniques for pole-shaped object extraction from MLS point clouds. While Section 1 highlights limitations of baseline methods—such as voxel-based pole extraction techniques and adaptive radius cylinder models—this subsection provides a qualitative comparison to contextualize our method's advantages.

Unlike voxel-based methods, which rely on computationally intensive grid construction and transformations, our approach uses trajectory data to segment point clouds directly, simplifying preprocessing and reducing computational overhead. Similarly, adaptive radius cylinder models (e.g.,[6]) often require iterative optimization and extensive parameter tuning, whereas our method employs straightforward geometric descriptors (linearity and verticality) to achieve rapid pole detection. A key strength of our algorithm is its independence from labeled training data, a common requirement in many baseline techniques that increases preparation time and resource demands. By contrast, our reliance on raw 3D point clouds and trajectory information enables efficient processing on standard hardware, as demonstrated across diverse urban and suburban datasets (Section 2).

While baseline methods may offer robustness in specific scenarios, such as handling complex occlusions or varying object densities, they typically incur higher computational costs and preparatory complexity. Our approach prioritizes speed and simplicity, achieving high performance (e.g., 97.21% average precision, Table 1 without sacrificing accuracy. This efficiency is particularly valuable for real-time or resource-constrained applications, distinguishing our method from existing techniques that often trade off speed for additional data dependencies or processing steps. Future work could explore quantitative benchmarks against these baselines to further validate these advantages, though the current results underscore the practical benefits of our streamlined design.

4.5 Analysis of Misclassified Cases

While the proposed algorithm demonstrated robust performance across diverse datasets, achieving an average precision of 97.21% and recall of 96.01% Table 1, a small number of misclassified cases warrant critical discussion to understand limitations and guide future improvements. Notably, the algorithm missed two pole-shaped objects in the experimental evaluation, as observed in the suburban USA dataset (Section 3.1), where 796 points were undetected out of 33,490 true pole points.

Several factors may contribute to these misses. First, occlusion by dense vegetation or overlapping infrastructure (e.g., power lines or traffic signs) likely obscured the geometric signatures of these poles. In urban and suburban environments, poles are often situated near trees or cables, which can disrupt the continuity of point cloud data, reducing the effectiveness of linearity and verticality descriptors (Section 2.2). For instance, a pole partially occluded by foliage may exhibit insufficient vertical point density within the 0.9 m spherical neighborhood, failing to meet the DBSCAN clustering criteria (ϵ = 0.5 m, MinPts = 10). Second, poles with non-standard configurations—such as those with attached equipment (e.g., transformers) or irregular bases—may deviate from the expected geometric profile, leading to their exclusion during the ground clearance (0.5 m) or statistical outlier removal stages (Section 2.1.2). The suburban USA dataset, with its mix of trees, power lines, and varied pole designs, presented such challenges, as noted in Section 3.1.

These misclassifications have implications for the algorithm's applicability. In scenarios requiring exhaustive pole detection (e.g., autonomous navigation), missing even a small fraction of poles could impact reliability. However, the low false negative rate (e.g., 796 undetected points versus 33,490 detected) suggests that the algorithm remains highly effective for most practical purposes, such as infrastructure mapping, where near-complete detection suffices. The trade-off between sensitivity and specificity is evident here: stricter geometric and clustering criteria enhance precision (97.21%) by rejecting ambiguous cases but may overlook edge cases like the two missed poles.

To address these limitations, future enhancements could include adaptive neighborhood sizing based on local point density to better handle occlusions, or the integration of contextual features (e.g., pole-top equipment recognition) to improve detection of non-standard poles. Additionally, incorporating multi-pass clustering or post-processing to recover missed candidates could mitigate false negatives, though at the cost of increased computational complexity. These insights, derived from analyzing the misclassified cases, underscore the algorithm's strengths in typical scenarios while highlighting areas for refinement in challenging environments.

5 Conclusion and Future Works

The paper introduces an innovative algorithm for extracting pole-shaped objects from lidar point clouds, addressing the challenges of large-volume spatial data processing in urban and suburban environments. By systematically partitioning point cloud data and employing geometric features like verticality and linearity, the method efficiently identifies and clusters pole-like objects. The algorithm's robustness is demonstrated through extensive testing on complex datasets totaling 1.3 kilometers and 80 million points, achieving an impressive average F1 score of 96.61 percent. Its key advantages include simplicity of implementation, rapid extraction time, independence from training data, and compatibility with various MLS systems. While the results are promising, the authors suggest further validation across different lidar platforms such as airborne or terrestrial laser scanners to comprehensively assess the algorithm's adaptability and performance in diverse environmental conditions.

Acknowledgment

Authors would like to thanks Prof. Haiyan Guan and Wayne A Sarasua for sharing their datasets. Dr. Malihi is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sk lodowska-Curie grant agreement No. 101034337.

References

[1] Quan Zhang, Yuhang Dai, Tisheng Zhang, Chi Guo, and Xiaoji Niu. Road semantic-enhanced land vehicle integrated navigation in gnss denied environments. *IEEE Transactions on Intelligent Transportation Systems*, 25(12):20889–20899, 2024. doi:10.1109/TITS.2024.3449892.

[2] Danesh Shokri, Heidar Rastiveis, Seyed Mohammad Sheikholeslami, Reza Shahhoseini, and Jonathan Li. Fast extraction of power lines from mobile lidar point clouds based on svm classification in non-urban area. *Earth Observation and Geomatics Engineering*, 5(2):63–73, 2021. ISSN 2588-4352. doi:10.22059/eoge.2022.317348.1092.

[3] Yaowen Pei, Feng Chen, Tao Ma, and Gonghui Gu. A comparative review study on the electrified road structures: Performances, sustainability, and prospects. *Structures*, 62:106185, 2024. ISSN 2352-0124. doi:10.1016/j.istruc.2024.106185. URL `https://www.sciencedirect.com/science/article/pii/S2352012424003370`.

[4] Inam Ullah, Deepak Adhikari, Habib Khan, M. Shahid Anwar, Shabir Ahmad, and Xiaoshan Bai. Mobile robot localization: Current challenges and future prospective. *Computer Science Review*, 53:100651, 2024. ISSN 1574-0137. doi:10.1016/j.cosrev.2024.100651. URL `https://www.sciencedirect.com/science/article/pii/S1574013724000352`.

[5] S. Malihi, M. J. Valadan Zoej, M. Hahn, M. Mokhtarzade, and H. Arefi. 3d building reconstruction using dense photogrammetric point cloud. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3:71–74, 2016. doi:10.5194/isprs-archives-XLI-B3-71-2016. URL `https://isprs-archives.copernicus.org/articles/XLI-B3/71/2016/`.

[6] Yueqian Shen, Junjun Huang, Jinguo Wang, Jundi Jiang, Junxi Li, and Vagner Ferreira. A review and future directions of techniques for extracting powerlines and pylons from lidar point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 132:104056, 2024. ISSN 1569-8432. doi:10.1016/j.jag.2024.104056. URL `https://www.sciencedirect.com/science/article/pii/S1569843224004102`.

[7] D. Shokri, H. Rastiveis, A. Shams, and W. A. Sarasua. Utility poles extraction from mobile lidar data in urban area based on density information. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W18:1001–1007, 2019. doi:10.5194/isprs-archives-XLII-4-W18-1001-2019. URL `https://isprs-archives.copernicus.org/articles/XLII-4-W18/1001/2019/`.

[8] Haris Balta, Jasmin Velagic, Walter Bosschaerts, Geert De Cubber, and Bruno Siciliano. Fast statistical outlier removal based method for large 3D point clouds of outdoor environments. *IFAC-PapersOnLine*, 51(22):348–353, 2018. ISSN 2405-8963. doi:10.1016/j.ifacol.2018.11.566. 12th IFAC Symposium on Robot Control SYROCO 2018.

[9] Wuming Zhang, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. An easy-to-use airborne lidar data filtering method based on cloth simulation. *Remote Sensing*, 8(6):501, 2016. ISSN 2072-4292. doi:10.3390/rs8060501. URL `https://www.mdpi.com/2072-4292/8/6/501`.

[10] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and S. Sarasvady. Dbscan: Past, present and future. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 232–238, 2014. doi:10.1109/ICADIWT.2014.6814687.