Spatio-Temporal Detection and Filtering of Dynamic Objects in Mobile LiDAR Point Clouds

Mustafa Zeybek[1]

[1] Selçuk University, Güneysınır Vocational School, 42490, Konya, Türkiye - mzeybek@selcuk.edu.tr

Keywords: Mobile LiDAR,Dynamic Object Removal, Point Cloud Processing, Ground Classification, Clustering Algorithms, Timestamp Analysis.

Abstract:

Mobile LiDAR systems are increasingly utilized for high-precision mapping in dynamic environments, yet the presence of moving objects introduces significant noise and distortions in the resulting point clouds. Addressing this challenge, this study proposes a novel and efficient method for detecting and removing moving objects from mobile LiDAR point clouds. The approach involves an initial separation of ground and non-ground points using the Cloth Simulation Filtering (CSF) algorithm, followed by density-based clustering (DBSCAN) of non-ground points. By analyzing the temporal distribution of LiDAR points (gpstime) within each cluster relative to ground points, clusters are classified as either static or dynamic. Dynamic clusters, corresponding to moving objects, are then excluded from the dataset, yielding a refined point cloud that better represents the static environment. The method is implemented in R using various open-source libraries and validated on high-traffic urban datasets acquired with the Riegl VMX-450 mobile LiDAR system. Experimental results demonstrate that the proposed pipeline effectively detects and removes dynamic objects, thereby improving the accuracy and reliability of LiDAR-based mapping in complex, real-world scenarios.

1. INTRODUCTION

In recent years, mobile LiDAR (Light Detection and Ranging) systems have become a cornerstone of 3D environmental mapping, supporting a wide array of applications such as autonomous driving, terrain modeling, city planning, and environmental assessment (Fang, 2014). These systems, equipped with laser sensors on mobile platforms, can scan their surroundings with high precision, collecting spatial coordinates and timestamps for each data point. This results in the rapid generation of detailed 3D point clouds. The efficiency and accuracy of mobile LiDAR make it a superior alternative to traditional survey methods, especially in dense or difficult-to-navigate urban landscapes.

However, significant challenges remain in the processing of data collected via mobile LiDAR. Chief among these is the presence of dynamic points caused by moving objects (e.g., vehicles, pedestrians, bicycles) encountered during data acquisition. These dynamic points introduce noise and distortions into the point cloud, potentially leading to errors in mapping and analysis processes. This issue is particularly pronounced in urban environments with high traffic density and pedestrian movement, where dynamic points can substantially degrade overall data quality, resulting in misclassification, segmentation errors, or inaccurate modeling outcomes.

In the literature, various methods have been developed for the detection and separation of dynamic objects in mobile LiDAR data. In particular, Simultaneous Localization and Mapping (SLAM)-based systems are widely employed in both academic research and industrial applications (Peng et al., 2024). However, SLAM-based approaches often suffer from reduced mapping accuracy in highly dynamic environments due to artifacts caused by moving objects. In addition, the literature includes scan-comparison-based approaches and occupancy grid-based methods for detecting dynamic points (Krishtopik and Yudin, 2023, Xiao et al., 2017). These methods rely on analyzing differences between consecutive scans or on grid-based mapping of the point cloud (Schauer and Nuchter, 2018). Nevertheless, such approaches generally require multiple overlapping scans and are associated with high computational costs (Habibiroudkenar et al., 2024).

While techniques like DBSCAN (Ester et al., 1996) offer strong clustering capabilities, they have rarely been combined with ground segmentation and timestamp-based analysis in mobile LiDAR workflows. In this study, we introduce a comprehensive approach that brings these techniques together to automatically detect and remove dynamic points—particularly from road surfaces—within point cloud datasets. By integrating ground classification, density-based clustering, and temporal filtering, our method delivers significant gains in detection precision and computational efficiency. This leads to point clouds that are cleaner and more suitable for tasks such as mapping and urban modeling, with test results showing reliable performance across various complex environments.

2. CHARACTERISTICS OF MOBILE LIDAR POINT CLOUDS

Mobile LiDAR systems generate high-density and irregular point clouds that represent the spatial position and physical properties of target surfaces within a unified reference coordinate system. Each point is typically recorded in the standard LAS format, which includes three-dimensional coordinate data (X, Y, Z), timestamp (GPS time), return intensity, classification label, number of echoes, scan angle, and, if available, RGB color values. LAS is the industry-standard format commonly used for storing and processing LiDAR data.

In mobile LiDAR, laser pulses are emitted from a moving platform, and the positions of points are determined by detecting

the returning signals. A single laser pulse may generate multiple returns, especially in complex urban environments. The first returns generally correspond to the top surfaces of objects, while later echoes may come from lower layers or background objects. This multi-return capability provides significant advantages in analyzing layered structures such as vegetation and buildings.

Since point clouds do not form a continuous mathematical surface, relationships between points are typically defined using geometric and attribute-based methods. Consequently, the classification and filtering of mobile LiDAR data are often based on features such as geometric descriptors, return intensity, and the number of echoes. The resulting datasets are widely applied in various domains, including urban modeling, infrastructure inventory, dynamic object detection, and mapping.

3. METHODOLOGY

3.1 Data Attributes and Preprocessing

Mobile LiDAR systems typically produce raw point clouds in which each point is structured to include spatial coordinates (x, y, z) and the corresponding measurement time, commonly referred to as `gpstime`. This structure allows for the acquisition of a rich, multi-dimensional dataset with both spatial and temporal information.

The first step in processing raw LiDAR data involves data loading and preprocessing. During the preprocessing stage, the following operations are typically performed:

- **Data Import:** Point cloud data in LAS/LAZ format is loaded into memory using tools such as *lidR* (Roussel and Auty, 2025), *PDAL* (PDAL Contributors, 2023), or *LAStools* (Isenburg, 2023).
- **Noise and Duplicate Filtering:** Repeated points and outliers—often resulting from sensor inaccuracies or environmental interference—are removed using established filtering techniques (Rusu and Cousins, 2011, Zhang et al., 2016).
- **Coordinate Transformation:** The point cloud's coordinate reference system is converted, if needed, to match the spatial framework of the study area.
- **ROI Extraction:** A subset of the data, corresponding to the region of interest, is extracted to minimize unnecessary computational load.

Effective preprocessing is fundamental to LiDAR data processing pipelines, given the frequent presence of spurious points caused by sensor-induced noise, GNSS errors, and transient objects. If such noise is not eliminated, it can propagate significant inaccuracies into subsequent tasks such as segmentation, object classification, and spatial analysis (Vosselman, 2004).

A key component of preprocessing involves the preliminary separation of ground and non-ground points. Common methods at this stage include height-based thresholding, morphological operations, and advanced filtering techniques such as Cloth Simulation Filtering (CSF) (Zhang et al., 2016). Return intensity (`intensity`) and related attributes may offer valuable information for the detection of noise, particularly in multi-return datasets.

In this study, the preprocessing phase—including data import and cleaning—was performed rigorously. Both spatial and temporal attributes of each LiDAR point were considered to prepare a high-quality dataset for the subsequent analysis stages, including density-based clustering and temporal segmentation.

3.2 Ground and Non-Ground Separation

Ground segmentation was performed in the first stage using the Cloth Simulation Filtering (CSF) algorithm (Zhang et al., 2016), a robust method recognized in the literature for separating ground surfaces from buildings and vegetation in complex LiDAR point clouds. CSF operates by inverting the point cloud and simulating the effect of gravity on a virtual cloth layer. As the cloth settles onto the lowest sections of the inverted surface, it conforms to the terrain, allowing the ground surface to be approximated. Subsequently, the original point cloud is thresholded against this surface to classify ground and non-ground points. Sınıflandırma süreci temel olarak aşağıdaki şekilde ifade edilebilir:

$$G = p \in P \mid \text{isGround}(p) = \text{True}, \quad NG = P \setminus G \tag{1}$$

In this formulation, P represents the full set of LiDAR points, G denotes the identified ground points, and NG refers to the remaining non-ground points, including buildings, vegetation, and vehicles.

The CSF algorithm is particularly advantageous due to its sensitivity to terrain variability, scalability to large datasets, and tunable parameters that allow for application across diverse topographies. Its efficacy has been widely demonstrated in urban, forest, and complex natural environments (Sarıtaş and Kaplan, 2023). In this study, CSF was employed to isolate ground points, thereby producing a refined data subset optimized for the detection of dynamic objects in the point cloud.

3.3 Clustering Non-Ground Points

The non-ground points (NG) were clustered using the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm (Ester et al., 1996) to detect scene objects and potentially dynamic entities. DBSCAN is frequently applied in LiDAR data processing due to its capability to handle complex and irregular spatial structures. It can automatically identify dense regions (clusters) in the data while effectively separating sparse or irrelevant points as noise (Ma et al., 2019).

The core principle of DBSCAN is to group together points that are within a specified distance threshold (ϵ) and have at least a minimum number of neighbors ($minPts$). This enables the identification of high-density areas as meaningful clusters, whereas boundary or sparsely distributed points are marked as noise.

In this step, the clustering of non-ground points can be expressed as follows:

$$C = \text{cluster}(NG), \quad C = \{C_1, C_2, \ldots, C_n\} \tag{2}$$

Here, C denotes the set of all clusters, and each C_i potentially represents an object or object group within the scene.

The main advantages of using the DBSCAN algorithm on LiDAR point clouds are as follows:

- It does not require the number of clusters to be specified in advance; clusters are identified automatically.
- It can successfully distinguish objects with varying densities and irregular distributions.
- Noise and outlier points can be effectively detected and excluded, leading to more accurate analysis results.

Density-based clustering algorithms, such as DBSCAN, have been widely employed for the segmentation and automatic extraction of urban objects including vehicles, pedestrians, and trees (Ferrara et al., 2018). In this study, DBSCAN was applied to non-ground points to identify candidate object clusters, each potentially representing a distinct scene entity. These clusters were subsequently utilized for temporal analysis aimed at dynamic object detection.

3.4 Timestamp-Based Dynamic Point Classification

In LiDAR systems, GPS time is assigned to each LiDAR return by appending a timestamp to the signal data. The internal clock of each LiDAR sensor is synchronized with a GPS time source to ensure temporal accuracy.

In this study, for each cluster C_i in the point cloud, the distribution of the timestamp attribute (`gpstime`) is analyzed and compared with that of the ground points located within the same 2D spatial boundary. This comparison plays a critical role in determining whether the object represented by the cluster is static or dynamic within the scene.

Points reflected from static objects are generally captured within a time interval similar to that of surrounding ground points. In contrast, points from moving (dynamic) objects tend to be concentrated within a shorter or distinct time interval. The core innovation of the proposed method lies in exploiting this temporal discrepancy to distinguish dynamic objects from static ones. The core of this analysis is based on the temporal difference between the cluster and the ground points, which is formulated as follows:

$$\Delta t_i = |\max(\text{gpstime}(C_i)) - \min(\text{gpstime}(G))| \quad (3)$$

Where:

- $\max(\text{gpstime}(C_i))$: the maximum timestamp among the points in cluster C_i,
- $\min(\text{gpstime}(G))$: the minimum timestamp among the ground points within the same spatial boundary.

Thus, Δt_i represents the temporal deviation between the cluster and the surrounding ground points.

The classification of clusters is performed as follows:

- **Static:** If the value of Δt_i is below a predefined threshold and the time distribution of the cluster resembles that of the ground points, the cluster is classified as **static**.
- **Dynamic:** Otherwise, the cluster is considered **dynamic** (i.e., belonging to a moving object).

Clusters identified as dynamic are removed from the dataset to obtain a cleaner point cloud that represents only the static environment. This can be expressed mathematically as:

$$P_{\text{static}} = P \setminus \{p \in C_i \mid \text{dynamic}(C_i) = \text{True}\} \quad (4)$$

At this stage, the timestamp-based analysis performed for each cluster enables the automatic and accurate removal of dynamic objects from the point cloud. As a result, a refined dataset is obtained that exclusively represents static elements in the scene.

3.5 Algorithmic Implementation

The proposed pipeline in this study consists of two main algorithmic components. The first algorithm covers the preprocessing stage, which includes clustering and spatial alignment operations. The second algorithm represents the classification stage, where cluster classification is performed based on timestamp density analysis.

In the preprocessing stage, the raw point cloud data is loaded, duplicate and noisy points are removed, and ground and non-ground points are separated. Subsequently, clusters are generated over the non-ground points using the DBSCAN algorithm, and the spatial boundaries of each cluster are identified and aligned accordingly. These steps ensure the extraction of meaningful candidate objects for analysis and enhance the accuracy of subsequent processing.

In the classification stage, the timestamp (`gpstime`) distribution of each cluster is analyzed. Based on the distribution pattern of measurement times within a cluster, it is automatically determined whether the cluster corresponds to a static (stationary) or dynamic (moving object) entity.

All processing steps were carried out within the `R` programming environment (R Core Team, 2025), utilizing open-source and robust packages for data loading, processing, clustering, and visualization tasks.

The main R packages used in this study are listed in Table 1.

Through the use of these packages, it was possible to efficiently process large-scale LiDAR datasets and present the results in a clear and visually rich manner. Thanks to the flexibility of the R environment, the proposed workflow can be easily adapted to different datasets and offers a practical solution for real-world applications.

The algorithm for the detection and removal of moving objects consists of two main steps: the first is pre-processing (Algorithm 1), and the second is filtering dynamic points (Algorithm 2).

The presented algorithm performs spatio-temporal classification of clusters within a mobile LiDAR point cloud using the `gpstime` (GPS timestamp) attribute. For each detected non-ground cluster, it first extracts the associated ground and non-ground points, merges them, and sorts the combined set by their

Package	Description
`lidR`	Provides comprehensive functions for LiDAR data processing, filtering, and analysis.
`sf`	Enables manipulation and analysis of vector-based spatial data.
`dplyr`	Facilitates data manipulation and transformation tasks.
`ggplot2`	Offers advanced and customizable data visualizations.
`plotly`	Supports the creation of interactive and dynamic plots.
`dbscan`	Implements density-based clustering algorithms for data segmentation.
`concaveman`	Generates concave polygons from point clusters.
`rlas`	Reads and writes LiDAR data in LAS format.
`RCSF`	Applies the Cloth Simulation Filtering (CSF) algorithm for ground point extraction.
`zoo`	Provides tools for time series data analysis and management.
`pracma`	Includes mathematical and statistical analysis functions.
`data.table`	Offers fast and efficient data processing for large datasets.
`caret`	Facilitates training, evaluation, and comparison of machine learning algorithms.
`pROC`	Provides tools for drawing and analyzing ROC curves.

Table 1. Main R packages used in the study and their functionalities. All packages are available at `https://cran.r-project.org/`.

Algorithm 1 Mobile LiDAR Pre-Processing Pipeline

1: Set `lidR` threads to maximum available (`set_lidr_threads(0)`)
2: Load LAS data; if EPSG missing, assign EPSG:5255
3: **if** `RCSF` package available **then**
4: Remove duplicates
5: Classify noise using `ivf(5,5)`
6: Classify ground using CSF (`sloop_smooth=TRUE`, `cloth_resolution=1`, `time_step=0.65`, `rigidness=3`)
7: **end if**
8: Extract non-ground points (`Classification==0`) and ground points (`Classification==2`)
9: Apply DBSCAN to non-ground (`eps=0.5`, `minPts=10`); keep clusters $\geq$10 points
10: Save `nonground.las` and `ground.las`
11: Compute concave hulls for each valid cluster (`ratio=0.05`)
12: Clip ground points by cluster boundaries and assign `ClusterID`
13: Save `clipped_ground_clusters.las`

timestamps. It then computes rolling statistics (mean and variance) and estimates density curves for both point types to analyze their temporal distribution. By detecting the number of peaks and comparing the time span ranges of ground (`g_range`) and non-ground (`ng_range`) points, the algorithm applies a rule-based decision system: clusters are classified as dynamic if their time span is narrower and peak count is lower than the surrounding ground; as static if both have similar peak structures; or as unclassified otherwise. The results are saved in CSV

Algorithm 2 Cluster Classification via `gpstime`

for each `ClusterID` in non-ground clusters **do**
  Extract corresponding ground and non-ground points
  Merge; sort by `gpstime`
  Compute `rolling_mean(k=4)` and `rolling_var(width=4)` per type
  Estimate density curves for ground and non-ground
  Detect peaks using `npeaks=4` for each type
  Calculate time spans: g_range and ng_range
  if $ng\_range < g\_range$ and $gp > ngp$ **then**
    Classify as Dynamic Non-Ground
  else if $gp = ngp$ and $g\_range > ng\_range$ **then**
    Classify as Dynamic Non-Ground
  else if $gp \geq 2$ and $ngp \geq 2$ **then**
    Classify as Static Non-Ground
  else
    Classify as Unclassified
  end if
end for
Save cluster classification results to CSV

format for further processing or visualization, enabling effective filtering of moving objects from the scene.

Step	Parameter	Value / Description
1	EPSG Code Assignment	5255 (TURKEY_ITRF96)
1	Noise Filtering	`ivf`(5, 5)
1	Ground Classification Method	Cloth Simulation Filtering (CSF)
1	CSF: Slope Smooth	TRUE
1	CSF: Class Threshold	0.1
1	CSF: Cloth Resolution	1
1	CSF: Time Step	0.65
1	CSF: Rigidness	3
2	DBSCAN `eps`	0.5 meters
2	DBSCAN `minPts`	10 points
2	Cluster Filtering	Clusters with size $\geq$ 10 points
3	Concave Hull Ratio	0.05 (5%)
4	gpstime Rolling Mean Window	4 points
4	gpstime Rolling Variance Window	4 points
4	Density Peak Detection	`npeaks` = 4
4	Classification Decision Rule	Based on time span comparison (g_range, ng_range) and peak counts (gp, ngp)
5	LAS Update Classification Labels	Class 3 = Static Non-Ground, Class 4 = Dynamic Non-Ground

Table 2. Parameters Used in the Classification Pipeline by Step Number

The parameters used in the implementation of the proposed methods are presented in Table 2.

3.6 Accuracy Metrics for Point Cloud Classification

To quantitatively evaluate the performance of the dynamic object detection and classification algorithm, several standard accuracy metrics commonly used in the literature were adopted. These include:

- **Overall Accuracy (OA):** The ratio of correctly classified points (both static and dynamic) to the total number of points. It provides a general indication of classification performance.
- **Precision:** Indicates how many of the points predicted as dynamic are actually dynamic. It is calculated as the ratio of true positives (TP) to all predicted positives (TP + FP).
- **Recall:** Measures how many of the actual dynamic points were correctly detected by the algorithm. It is the ratio of true positives (TP) to all actual positives (TP + FN).
- **F1-Score:** The harmonic mean of precision and recall; it provides a balanced measure that considers both false positives and false negatives.
- **Intersection over Union (IoU):** The ratio of the intersection to the union between the predicted and actual dynamic (or static) point sets. It is frequently used for segmentation performance assessment.

The following formulas (5–9) were used for the computation of these metrics:

$$\text{OA} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{9}$$

Where:

- TP (*True Positive*): Points correctly classified as dynamic,
- TN (*True Negative*): Points correctly classified as static,
- FP (*False Positive*): Static points incorrectly classified as dynamic,
- FN (*False Negative*): Dynamic points incorrectly classified as static.

These metrics allow for a quantitative assessment of the algorithm's ability to both accurately detect dynamic objects and preserve static scene elements. In particular, metrics such as F1-Score and IoU provide a more comprehensive evaluation in cases of class imbalance and during object segmentation tasks.

4. CASE STUDY

4.1 Study Area and Application Scenario

In this study, to evaluate the performance of the proposed algorithm, a test site was selected in the city center of Konya, Türkiye—an area representative of typical urban morphology. The selected test area covers an urban region approximately 70 meters in length and 3776 m^2 in size, and includes a variety of object types and urban elements (Figure 1). The study area meets the requirements of a typical urban mapping application in terms of object diversity, moving object density, and land use variation. Since the analysis primarily focuses on road surfaces, the presence of moving vehicles in the dataset was particularly important.

As part of the project, the data collected from the field enabled the creation of a digital model of the roadway and surrounding urban surface. In particular, the study provided a valuable framework for evaluating advanced filtering techniques aimed at road surface analysis and integration with urban Geographic Information Systems (GIS).

Figure 1. a) Location of the study area on the map (Leaflet) and the raw point cloud data visualized in grayscale based on intensity values.

A summary of the LiDAR data specifications used in this study is presented in Table 3.

Property	Value
File Format	LAS v1.2 (Point Data Format: 1)
Coordinate System	TUREF / TM33 (EPSG:5255)
Area	3,776 m^2
Total Number of Points	5,198,227
Point Density	1,376.65 points/m^2
Pulse Density	1,349.38 pulses/m^2
Scanning Type	Terrestrial (Mobile)
Height (Z) Range	1082.99 m – 1107.341 m
X Coordinate Range	457524.6 – 457586.5
Y Coordinate Range	4205192 – 4205269
GPS Time Type	GPS Week Time
Scale Factors	X: 0.00025, Y: 0.00025, Z: 0.00025
Offset Values	X: 457543, Y: 4205208, Z: 0
Number of Returns	1st: 5,095,277 2nd: 94,813 3rd: 7,848 4th: 289
Date of Data Creation	Day 211 / Year 2025
Generating Software	LASzip DLL 3.4 r3 (191111)

Table 3. Basic Properties of the LAS Dataset Used

4.2 3D Measurement and Data Acquisition Process

A mobile mapping system (MMS) equipped with a 3D LiDAR sensor and integrated camera was deployed on a vehicle to enable efficient acquisition of high-density point cloud data at urban-compliant speeds (20–30 km/h). Each LiDAR point included spatial coordinates along with attributes such as timestamp (`gpstime`) and return intensity. The proposed dynamic object detection algorithm was applied to this dataset through a three-stage process: ground filtering, density-based clustering, and timestamp-based classification. This workflow produced a refined point cloud representing only static urban elements, suitable for applications in digital mapping, urban planning, and infrastructure management.

The classification of points in the LAS (LASer Format) files follows the standard class codes defined by the ASPRS (American Society for Photogrammetry and Remote Sensing). However, there is no predefined standard class code for "dynamic objects". For the purposes of this study, static objects were labeled as `Class 3`, and dynamic objects as `Class 4`, to ensure clarity and ease of processing.

4.2.1 RIEGL VMX-450 Mobile LiDAR System The RIEGL VMX-450 is an advanced mobile mapping system with high precision and fast data acquisition capabilities. It is specifically designed for capturing detailed three-dimensional (3D) point cloud and image data in large and complex environments such as urban areas, highways, railways, and major infrastructure projects. Further details about the system are provided in Table 4.

This system was chosen for the study due to its ability to generate dense and high-quality point cloud data efficiently.

Feature	Description
Dual LiDAR Scanners	Two RIEGL VQ-450 laser scanners enabling wide scan angle and high-density data collection
High Scan Rate	Up to 1.1 million points per second
Positioning	Integrated GNSS/IMU system for accurate georeferencing
Camera Integration	High-resolution RGB panoramic imaging
Modular Design	Easy mounting on various vehicles, quick field setup
Software Support	Compatible with RiPROCESS, RiPRECISION, and RiACQUIRE for processing and QA/QC

Table 4. Key Features of the RIEGL VMX-450 Mobile LiDAR System

5. RESULTS

The classification performance of the proposed method is summarized in Tables 5 and 6. The confusion matrix (Table 5) demonstrates a balanced and effective classification capability, with the model correctly identifying 85,285 static and 74,726 dynamic instances. Misclassifications were relatively limited, with 9,223 static points incorrectly labeled as dynamic and 7,185 dynamic points misclassified as static. As detailed in Table 6, both classes achieved high performance metrics. The static class obtained a precision of 92.23%, recall of 90.24%, and F1-score of 91.22%, while the dynamic class achieved a precision of 89.01%, recall of 91.23%, and F1-score of 90.11%. Additionally, the Intersection over Union (IoU) values reached 83.87% for static and 82.00% for dynamic classes, indicating strong spatial reliability. The overall classification accuracy of 90.70% further underscores the robustness and generalizability of the proposed approach.

Actual / Predicted	Static	Dynamic
Static	85285	9223
Dynamic	7185	74726

Table 5. Confusion Matrix of Classification Results

Metric	Static Class (%)	Dynamic Class (%)
Precision	92.23	89.01
Recall	90.24	91.23
F1-Score	91.22	90.11
IoU	83.87	82.00
Overall Accuracy	90.70 %	

Table 6. Classification accuracy metrics for the proposed method

The results clearly indicate that the model performs in a well-balanced and reliable manner across both static and dynamic categories. The relatively high IoU values further reflect the spatial coherence of the predictions, which is particularly commendable given the inherent challenges of time-dependent point cloud classification. Achieving over 90% performance reinforces the potential of the proposed method for real-world deployment scenarios and practical applications in dynamic scene understanding.

Figure 2 illustrates the three-stage classification pipeline applied to LiDAR point cloud data. In Figure 2a, the ground extraction step is shown. Points classified as ground (Class = 2) are visualized in blue, while non-ground points (Class = 1) are rendered in grayscale. The Cloth Simulation Filtering (CSF) method successfully distinguished ground surfaces from elevated objects. Notably, features such as trees, vehicles, lamp posts, and the overpass are clearly identified as non-ground elements.

Figure 2b presents the DBSCAN clustering results, where non-ground points are segmented into discrete clusters, each denoted by a unique `ClusterID` and represented in distinct colors. Ground points remain visible in the background in grayscale for contextual reference. The DBSCAN algorithm effectively grouped semantically meaningful objects such as trees, vehicles, traffic signs, lamp posts, and the overpass into separate clusters. Noise and minor irrelevant components were filtered out during post-processing.

Finally, Figure 2c shows the dynamic vs. static object classification. Here, points are assigned to the following classes based on the `Classification` attribute: 1 — Unclassified/Other, 2 — Ground, 3 — Static Non-Ground (e.g., fixed structures), and 4 — Dynamic Non-Ground (e.g., moving vehicles, pedestrians). Trees and infrastructure elements such as lamp posts and signs were predominantly classified as static, while vehicles were accurately identified as dynamic. The timestamp-based density and variance analysis employed in this final stage proved highly effective in distinguishing between dynamic and static entities.

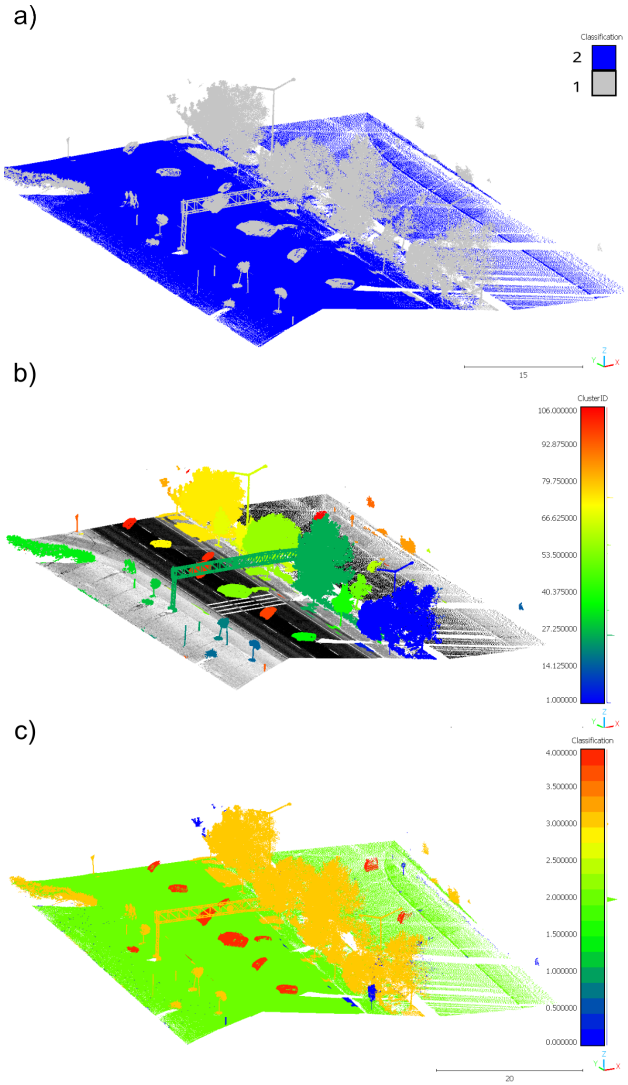

Figure 2. Three-stage classification workflow applied to LiDAR point cloud data. (a) Initial ground classification using the CSF (Cloth Simulation Filtering) algorithm. Blue indicates ground points (Class 2). (b) Object segmentation of non-ground points using the DBSCAN clustering algorithm. Clusters are color-coded based on their `ClusterID`, with each cluster representing a potential object group (e.g., trees, vehicles, lamp posts). (c) Final classification based on `gpstime`-based density and variance analysis. Green: ground (Class 2), red: dynamic objects (Class 3), orange: static objects (Class 4), blue: other/unclassified points.

6. DISCUSSION AND LIMITATIONS

The results highlight the robustness and practical applicability of the proposed method in challenging and dynamic urban scenarios. The integration of timestamp analysis with ground filtering and object clustering enables reliable detection of moving objects and achieves performance levels comparable to widely used occupancy grid and scan-matching techniques reported in the literature (Zeng et al., 2024).

The distinction between static and dynamic objects is based on the analysis of peak values in the timestamp (`gpstime`) density distributions. This approach assumes a regular movement pattern during scanning, and its generalizability may be limited in datasets with irregular trajectories or excessive scan overlap.

The classification process relies on fixed and pre-defined threshold values (e.g., number of peaks, time range comparison), which may need to be re-calibrated for different environments, sensor systems, or datasets. Each cluster is evaluated independently, without considering its spatial or temporal relationship with neighboring clusters. This may result in the loss of potentially valuable contextual information that could enhance the detection of dynamic objects.

The accuracy metrics were computed based on a manually labeled `.las` file serving as ground truth. However, the quality, completeness, and object diversity of this reference data should ideally be more detailed. Incomplete or imbalanced labeling may impact the reliability of the evaluation. Nevertheless, since the Balanced Accuracy exceeded 90%, it can be concluded that the model was not significantly affected by class imbalance in this particular study.

Clusters containing fewer than 10 points were excluded from the analysis. This design choice may lead to reduced sensitivity to small dynamic objects such as pedestrians or bicycles, which could be missed due to their smaller point cloud footprint.

The current study proposes a fully rule-based method and does not incorporate statistical modeling or deep learning approaches capable of learning complex patterns between dynamic and static objects. Parameters of the DBSCAN algorithm (such as `eps` and `minPts`), as well as those of the CSF ground filtering method, were manually tuned for this specific dataset. These parameters may not yield equivalent performance under different LiDAR resolutions or scanning speeds.

The implemented method operates as a batch-processing pipeline and is not yet designed for integration with real-time SLAM or live LiDAR streaming. However, the framework could theoretically be adapted for real-time analysis with appropriate architectural modifications.

Currently, the timestamp-based classification uses only peak count and distribution width. In future work, integrating more advanced time series analysis techniques—such as Fourier transforms, wavelet decomposition, or change-point detection—could improve robustness and interpretability.

Finally, the current rule-based decision trees used for dynamic/static classification could be extended with supervised machine learning algorithms or deep learning models to enable more generalizable and automated labeling systems (Zhang et al., 2020, Wang et al., 2024).

In future studies, adaptive parameter tuning based on point density, scene structure, and data variability should be explored to improve the flexibility and performance of algorithms like DBSCAN and CSF.

7. CONCLUSION

In this study, an integrated method has been proposed for detecting and removing moving objects from road surfaces in mobile LiDAR point clouds. The approach combines ground classification, density-based clustering (DBSCAN), and timestamp density analysis. The proposed workflow involves the separation of ground and non-ground points using the CSF algorithm, followed by clustering of non-ground points and temporal distribution analysis for each cluster.

Experimental results demonstrated that the method achieved high accuracy (90.70% OA) and balanced performance (F_1-score $> 90\%$, IoU $> 82\%$), indicating its effectiveness in filtering out dynamic objects. The use of GPS time (`gpstime`) as a feature proved to be a strong discriminator for dynamic/static classification, especially in dense and complex urban environments.

Future work will focus on enhancing the approach with adaptive parameter optimization, advanced time series analysis techniques, and supervised or deep learning-based classification methods. Additionally, integration with real-time mobile mapping systems will further expand the method's potential for field applications.

ACKNOWLEDGEMENTS

This research was supported by Koyuncu LiDAR Harita, whose data contributions and field expertise were essential to the success of this study. The author would also like to extend their sincere thanks to the technical team at Koyuncu LiDAR Harita for their valuable support and for providing the mobile LiDAR datasets used in this research.

REFERENCES

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, AAAI Press, 226–231.

Fang, X., 2014. Automatic detection of moving objects in mobile lidar point clouds. Master's thesis, University of Twente, Faculty of Geo-Information Science and Earth Observation, Enschede, The Netherlands. Supervisors: Dr. K. Khoshelham, Dr. ir. S.J. Oude Elberink.

Ferrara, R., Virdis, S. G., Ventura, A., Ghisu, T., Duce, P., Pellizzaro, G., 2018. An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN. *Agricultural and Forest Meteorology*, 262, 434-444. https://doi.org/10.1016/j.agrformet.2018.04.008.

Habibiroudkenar, P., Ojala, R., Tammi, K., 2024. DynaHull: Density-centric Dynamic Point Filtering in Point Clouds. *Journal of Intelligent & Robotic Systems*, 110, 165. https://doi.org/10.1007/s10846-024-02203-2.

Isenburg, M., 2023. LAStools - efficient lidar processing software. `https://rapidlasso.com/lastools/`. Accessed: 2025-08-06.

Krishtopik, A. S., Yudin, D. A., 2023. Monitoring of dynamic objects on a 2d occupancy map using neural networks and multimodal data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 48, International Society for Photogrammetry and Remote Sensing, 137–143.

Ma, Y., Anderson, J., Crouch, S., Shan, J., 2019. Moving Object Detection and Tracking with Doppler LiDAR. *Remote Sensing*, 11(10), 1154. https://doi.org/10.3390/rs11101154.

PDAL Contributors, 2023. Pdal: Point data abstraction library. `https://pdal.io`. Version 2.5.5.

Peng, H., Zhao, Z., Wang, L., 2024. A Review of Dynamic Object Filtering in SLAM Based on 3D LiDAR. *Sensors*, 24, 645. https://doi.org/10.3390/s24020645.

R Core Team, 2025. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Roussel, J.-R., Auty, D., 2025. Airborne LiDAR Data Manipulation and Visualization for Forestry Applications. R package version 4.2.1.

Rusu, R. B., Cousins, S., 2011. 3d is here: Point cloud library (pcl). *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1–4.

Sarıtaş, B., Kaplan, G., 2023. Enhancing Ground Point Extraction in Airborne LiDAR Point Cloud Data Using the CSF Filter Algorithm. *Advanced LiDAR*, 3(2), 53–61. http://publish.mersin.edu.tr/index.php/lidar/index.

Schauer, J., Nuchter, A., 2018. The Peopleremover-Removing Dynamic Objects from 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid. *IEEE Robotics and Automation Letters*, 3, 1679-1686. https://doi.org/10.1109/LRA.2018.2801797.

Vosselman, G., 2004. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46number 8/W2, ISPRS, 33–38.

Wang, X., Hu, X., Zhong, P., 2024. Visual reinforcement learning for dynamic object detection. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 48, International Society for Photogrammetry and Remote Sensing, 679–684.

Xiao, W., Vallet, B., Xiao, Y., Mills, J., Paparoditis, N., 2017. OCCUPANCY MODELLING FOR MOVING OBJECT DETECTION FROM LIDAR POINT CLOUDS: A COMPARATIVE STUDY. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4, 171-178. https://doi.org/10.5194/isprs-annals-IV-2-W4-171-2017.

Zeng, K., Shi, H., Lin, J., Li, S., Cheng, J., Wang, K., Li, Z., Yang, K., 2024. Mambamos: Lidar-based 3d moving object segmentation with motion-aware state space model. *Proceedings of the 32nd ACM International Conference on Multimedia*, ACM, 1505–1513.

Zhang, M., Fu, R., Guo, Y., Wang, L., 2020. Moving Object Classification Using 3D Point Cloud in Urban Traffic Environment. *Journal of Advanced Transportation*, 2020, 1-12. https://doi.org/10.1155/2020/1583129.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 501. https://doi.org/10.3390/rs8060501.