A Point Cloud Dataset of Vehicles Passing through a Toll Station for Use in Training Classification Algorithms

Alexander Campo-Ramírez *, Eduardo F. Caicedo-Bravo © and Eval B. Bacca-Cortes ©

School of Electrical and Electronic Engineering, Faculty of Engineering, Universidad del Valle, Cali 760032, Colombia; eduardo.caicedo@correo.univalle.edu.co (E.F.C.-B); bladimir.bacca@correo.univalle.edu.co (E.B.B.-C)
* Correspondence: campo.alexander@correo.univalle.edu.co

Abstract: This work presents a point cloud dataset of vehicles passing through a toll station in Colombia to be used to train artificial vision and computational intelligence algorithms. This article details the process of creating the dataset, covering initial data acquisition, range information preprocessing, point cloud validation, and vehicle labeling. Additionally, a detailed description of the structure and content of the dataset is provided, along with some potential applications of its use. The dataset consists of 36,026 total objects divided into 6 classes: 31,432 cars, campers, vans and 2-axle trucks with a single tire on the rear axle, 452 minibuses with a single tire on the rear axle, 1158 buses, 1179 2-axle small trucks, 797 2-axle large trucks, and 1008 trucks with 3 or more axles. The point clouds were captured using a LiDAR sensor and Doppler effect speed sensors. The dataset can be used to train and evaluate algorithms for range data processing, vehicle classification, vehicle counting, and traffic flow analysis. The dataset can also be used to develop new applications for intelligent transportation systems.

Dataset: The data presented in this study are openly available at: https://doi.org/10.5281/zenodo.10974361

Dataset License: Creative Commons Attribution 4.0 International

Keywords: dataset; point cloud; laser imaging detection and ranging (LiDAR); intelligent transportation systems (ITS); artificial vision; range data processing; vehicle classification

1. Introduction

Modern countries face an increasingly complex panorama: efficient traffic management and road safety. The rapid increase in the number of vehicles, traffic congestion, traffic accidents and environmental pollution are some of the problems that plague most countries [1,2]. These challenges not only negatively impact the quality of the life of citizens but also generate significant economic and environmental costs [3].

Responding to these urgent needs, intelligent transportation systems (ITS) emerge as an innovative technological solution that is changing the field of transportation [4]. These systems are a wide range of technological solutions applied to transportation, which integrate a network of subsystems and distributed sensors to collect data in real time about the road environment [5]. By processing this big data using machine learning algorithms and artificial intelligence techniques, ITS can obtain crucial information to optimize traffic flow; improve road safety and reduce the environmental impact of transportation [6].

ITS are fundamental to the continuous transformation of mobility and offer solutions to improve safety, efficiency and sustainability in transportation. At the core of these systems, one essential need is focused on real-time object detection and classification, especially as it relates to vehicles traversing roads and infrastructure points such as toll stations [7].
In recent years, the field of vehicle detection and classification has been revolutionized by the arrival of new sensor technologies and the increase in data processing algorithms. In particular, LiDAR (light detection and ranging) sensors have become a valuable tool for capturing high-resolution three-dimensional point cloud data, offering information on the dynamics of vehicular movement [8]. Processing these point clouds using specialized algorithms allows ITS to detect, track and recognize objects.

Accurate vehicle identification and classification are essential for numerous ITS applications, such as dynamic lane assignment, differentiated toll collection, traffic restrictions, implementation of automated tolling systems, emergency vehicle prioritization, and enforcement of traffic rules [9–12]. However, the development and validation of robust classification algorithms depend on access to high-quality datasets that faithfully represent real-world traffic scenarios. These datasets are the fundamental basis for training, testing and benchmarking many algorithms, allowing researchers to evaluate their performance under different environmental conditions and operational constraints [13].

In response to this demand, this work presents a novel point cloud dataset designed from the capture vehicles passing through a toll station in Colombia. Leveraging LiDAR and speed sensors, this dataset provides a detailed three-dimensional representation of vehicle geometry. Considering the richness of point cloud data, our dataset offers valuable insights into the various shapes and sizes of vehicles encountered in real-world traffic scenarios.

This article clarifies the methodologies used to collect, process and label the dataset, highlighting its suitability for training and validating classification algorithms adapted to vehicle recognition tasks in intelligent transportation systems. Additionally, this work mentions potential applications of this dataset, including but not limited to vehicle counting, size estimation, vehicle type classification and 3D modeling, thereby facilitating advances in traffic management, road safety, and overall efficiency of ITS solutions.

Figure 1 shows some examples of vehicle point clouds from the dataset; on the left are the raw 3D point clouds, and on the right are the 2D point clouds after processing to extract only the side profile of the vehicles, excluding background, ground surface and noise information. The point clouds are presented on Cartesian planes where the Z axis corresponds to the height of the vehicle, the X axis to the length of the vehicle and the Y axis to the width. The units of the Cartesian plane in the 2D point clouds for the X and Z axes are meters and allow the size of the vehicles to be observed and compared.

Figure 1. Dataset examples.

Technology is advancing rapidly, and with access to datasets like the one presented in this work, ITS are poised to revolutionize mobility. By sharing this dataset, the goal
is to foster collaboration between researchers to improve classification algorithms, drive innovation in ITS, and create safer and more efficient transportation networks for the future.

This paper is structured as follows: Section 1.1 reviews previous work related to vehicle point cloud datasets. Section 2 describes the structure and format of the dataset, along with a detailed description of the hardware used. Section 3 details the data acquisition process, including the processing stages and tools used for the construction and filtering of the point clouds from the dataset files. In addition, the methodology and tools used for labeling and validation are explained. Section 4 presents examples of applications of the data set. Finally, Section 5 presents the conclusions derived from this work.

1.1. Related Work

In recent years, the domain of ITS has witnessed remarkable progress in the development of vehicle detection and classification algorithms, fueled by advancements in sensor technologies, machine learning techniques and computational capabilities. These advancements have been instrumental in addressing the increasing demand for safer, more efficient, and sustainable transportation solutions [5]. In this context, point cloud datasets have become a fundamental tool for the development and evaluation of real-time vehicle classification algorithms.

Some of the widely used point cloud datasets for vehicle classification are the KITTI [14], Argoverse [15], nuScenes [16], Waymo Open [17], A2D2 [18], ApolloScape [19] and PandaSet [20] datasets, which feature traffic scenes captured from moving vehicles equipped with multiple sensors, including cameras and LiDAR. However, due to their generic nature, these datasets mainly focus on autonomous driving, which limits their direct applicability in specific scenarios such as the classification of vehicles traversing roads and infrastructure points such as toll stations.

Table 1 provides an overview of the characteristics of these datasets. The “Dataset” column specifies the name of the dataset, “Ann. fr.” indicates the number of annotation frames present and “3D box.” reflects the number of objects detected in the frames and represented by 3D point clouds. “LiDAR” indicates how many 3D LiDAR sensors were used to acquire the data, while “Classes” shows the number of categories into which the objects in the dataset have been labeled. The “Location” column indicates the city or cities where the data acquisition was performed, “Distance” indicates the distance traveled, “Night/Rain” indicates whether the data was taken under these conditions and, finally, “Duration” indicates the time duration of data collection.

The Waymo dataset stands out, containing 12 million identified objects in 230,000 frames. However, it only classifies these objects into four different categories. In contrast, NuScenes includes 1.4 million objects classified into 23 different categories. It is relevant to note that these categories cover a variety of objects, not only vehicles but also pedestrians, seated people, cyclists, trams and others.

Table 1. Overview of publicly available point cloud datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Year</th>
<th>Ann. Fr</th>
<th>3D Box.</th>
<th>LiDAR</th>
<th>Classes</th>
<th>Location</th>
<th>Distance</th>
<th>Night/Rain</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI [14]</td>
<td>2013</td>
<td>15 k</td>
<td>200 k</td>
<td>1</td>
<td>8</td>
<td>Karlsruhe</td>
<td>-</td>
<td>No/No</td>
<td>6 h</td>
</tr>
<tr>
<td>Argoverse [15]</td>
<td>2019</td>
<td>22 k</td>
<td>993 k</td>
<td>2</td>
<td>15</td>
<td>Pittsburgh, Miami</td>
<td>300 km</td>
<td>Yes/Yes</td>
<td>320 h</td>
</tr>
<tr>
<td>NuScenes [16]</td>
<td>2020</td>
<td>40 k</td>
<td>1.4 M</td>
<td>1</td>
<td>23</td>
<td>Boston, Singapore</td>
<td>242 km</td>
<td>Yes/Yes</td>
<td>15 h</td>
</tr>
<tr>
<td>Waymo [17]</td>
<td>2020</td>
<td>230 k</td>
<td>12 M</td>
<td>5</td>
<td>4</td>
<td>3 × USA</td>
<td>76 km²</td>
<td>Yes/Yes</td>
<td>6 h</td>
</tr>
<tr>
<td>A2D2 [18]</td>
<td>2020</td>
<td>12 k</td>
<td>-</td>
<td>5</td>
<td>14</td>
<td>3 × Germany</td>
<td>-</td>
<td>Yes/Yes</td>
<td>-</td>
</tr>
<tr>
<td>ApolloScape [19]</td>
<td>2020</td>
<td>144 k</td>
<td>70 k</td>
<td>2</td>
<td>35</td>
<td>4 × China</td>
<td>-</td>
<td>Yes/No</td>
<td>-</td>
</tr>
<tr>
<td>PandaSet [20]</td>
<td>2021</td>
<td>8 k</td>
<td>8 k</td>
<td>2</td>
<td>28</td>
<td>San Francisco</td>
<td>-</td>
<td>Yes/-</td>
<td>0.3 h</td>
</tr>
</tbody>
</table>
Although these datasets are valuable resources for research and development of perception systems in autonomous driving environments, they also present some specific limitations regarding vehicle classification [21]:

- **Vehicle variety:** Although these datasets contain a wide variety of vehicles common in urban environments, they may not fully represent the diversity of vehicle types found in other geographic regions. Examples of less common vehicles may be missing.

- **Resolution and data quality:** Although datasets typically have high resolution, data quality can vary depending on acquisition conditions, such as lighting, weather, or the presence of obstacles that may partially obstruct objects. This can make it difficult to accurately classify vehicles in some situations.

- **Point cloud information:** The density of point cloud data in sets such as these may be insufficient to provide a detailed description of each vehicle. The abundance of information in the point cloud depends on factors such as the distance between objects, weather conditions, and acquisition equipment. Consequently, vehicle point cloud data is often sparse, resulting in limited representation of different vehicle parts.

Considering these topics, several researchers have addressed these limitations by developing their datasets. These sets are generated from data acquisition systems strategically located at fixed points, both overhead and to the side of traffic routes. This strategy allows a wide variety of traffic scenarios to be captured from different perspectives, enriching the diversity and quality of data available for vehicle classification research.

For example, in Refs. [11,22], a dataset including 4955 vehicles was created using a LiDAR mounted overhead of a road. Five different classes were generated, ranging from motorcycles to trucks and buses. In Ref. [23], two LiDARs were installed on the side of a toll station road, generating a dataset with 206 vehicles for axle detection and counting in four different classes. In works [24,25], the authors placed three LiDARs on a gantry in a three-lane road, one scanner per lane, obtaining a dataset with 30,000 vehicles distributed in six different classes. Authors in Ref. [26] used two laser distance detectors on a gantry, one on each side of a gantry, along with a third sensor on a support pillar in front of the gantry, to collect data from 270 vehicles, including saloon cars, passenger cars and trucks. In Ref. [27], the authors collected data on an entrance ramp to a truck scale, using a LiDAR located on the side and collecting 10,024 vehicles in 11 different classes. Finally, the authors in [21] employed three LiDARs in a similar configuration to Ref. [26], achieving a dataset with 800 vehicles and high-density point clouds, covering 11 different vehicle types, from vans to fuel tank trucks.

Table 2 presents a summary of the main characteristics of these datasets, designed specifically for vehicle classification. The “Veh.” column reflects the size of the dataset, that is, the number of vehicles detected, “Type” indicates the type of laser sensors used, and “Sensor Position” specifies the location of the sensors on the road. The other columns are similar to Table 1. The work in Refs. [24,25] stands out, which has 30,000 available vehicles classified into six categories, including passenger vehicles, passenger vehicles with trailers, trucks, trucks with trailers, trucks with two trailers and motorcycles.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Veh.</th>
<th>LiDAR Type</th>
<th>Class.</th>
<th>Location</th>
<th>Sensor Position</th>
<th>Night/Rain</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdelbaki et al. [11,22]</td>
<td>2001</td>
<td>5 k</td>
<td>1 2D</td>
<td>5</td>
<td>USA</td>
<td>Overhead</td>
<td>-/Yes</td>
<td>-</td>
</tr>
<tr>
<td>Sato et al. [23]</td>
<td>2014</td>
<td>206</td>
<td>2 2D</td>
<td>4</td>
<td>Japan</td>
<td>Side</td>
<td>-/-</td>
<td>-</td>
</tr>
<tr>
<td>Chidlovskii et al. [24,25]</td>
<td>2013</td>
<td>30 k</td>
<td>3 2D</td>
<td>6</td>
<td>USA</td>
<td>Overhead</td>
<td>-/-</td>
<td>-</td>
</tr>
<tr>
<td>Xu et al. [26]</td>
<td>2015</td>
<td>270</td>
<td>3 2D</td>
<td>3</td>
<td>China</td>
<td>Side and overhead</td>
<td>-/-</td>
<td>-</td>
</tr>
<tr>
<td>Li et al. [27]</td>
<td>2023</td>
<td>10 k</td>
<td>1 3D</td>
<td>11</td>
<td>USA</td>
<td>Side</td>
<td>Yes/Yes</td>
<td>19 days</td>
</tr>
<tr>
<td>Ye et al. [21]</td>
<td>2023</td>
<td>800</td>
<td>3 2D</td>
<td>11</td>
<td>China</td>
<td>Side and overhead</td>
<td>-/-</td>
<td>-</td>
</tr>
<tr>
<td>This work</td>
<td>2024</td>
<td>30 k</td>
<td>1 2D</td>
<td>6</td>
<td>Colombia</td>
<td>Side</td>
<td>Yes/Yes</td>
<td>12 days</td>
</tr>
</tbody>
</table>
However, although these works show significant advances and important results in research on vehicle classification from point clouds, it is still possible to identify some limitations related to the data sets:

1. The data sets are not publicly accessible.
2. The number and variety of classes are limited. In Refs. [11,22,24,25], sensors mounted overhead of a road only allow classification into a few classes: motorcycle, passenger car, bus, and truck, with or without trailers. In Refs. [21,23,26], the number of objects is small, less than 1000. In Ref. [23], although the sensors were installed in a toll plaza, the same four vehicles were used for collection of data, one for each class, which does not reflect a real traffic environment. In Refs. [21,27], the focus was mainly on large vehicles.

In contrast, the main advantages of the data set presented in this work are as follows:

1. The data set is publicly accessible.
2. It contains a significant number of objects, 36,026, divided into 6 classes: 31,432 cars, campers, vans, and 2-axle trucks with a single tire on the rear axle; 452 minibuses with a single tire on the rear axle; 1158 buses; 1179 2-axle small trucks; 797 2-axle large trucks; and 1008 trucks with 3 or more axles.
3. The data were acquired in a real traffic environment 24 h a day for 12 days, allowing a wide variety of vehicles to be obtained. Although the data are labeled and classified into six classes according to Colombian regulations, it is possible to relabel it and find a wider range of classes according to specific needs. These classes include cars, pickups, campers or vans with trailer, 3-axle buses, single unit 3-axle trucks, single unit 4 or more-axle trucks, single trailer 3 or 4-axle trucks, single trailer 5-axle trucks, single trailer 6 or more-axle trucks, multi-trailer 5 or less-axle trucks, multi-trailer 6-axle trucks, multi-trailer 7 or more-axle trucks, barn trucks, fence trucks, crane trucks, semi-trailer tractors, garbage trucks, watering trucks, fuel tank trucks, among others.

As a result, while current datasets have been very useful for autonomous driving research, it is evident that there is a growing demand for specialized datasets designed for specific applications in ITS. These datasets are essential for training classification algorithms that can be applied in both ITS and autonomous driving tasks, capturing information relevant to the particular environment and specific tasks at hand. For example, a dataset focused on vehicles passing through a toll station would benefit from annotations that cover not only the type of vehicle, but also details such as the lane used, the traffic flow rate, and a more detailed description of the vehicles, which would facilitate a more precise classification and with a greater diversity of classes.

The dataset presented in this paper offers significant advantages compared to other works in terms of the number of vehicle samples and the diversity of vehicle classes. Consequently, it becomes a valuable tool for the development and research of intelligent transportation systems. This dataset has the potential to improve the accuracy of vehicle classification algorithms, as well as point cloud and range data processing, along with other crucial traffic analysis tasks.

2. Dataset Description

This section begins by offering a thorough description of the hardware used, covering its components, configuration, operating conditions and installation in a toll station in Colombia. This information lays the foundation for understanding the structure of the data. Next, a detailed description of the dataset is provided, including the files that comprise it and the format in which they are organized.

2.1. Hardware Description

The dataset was obtained using a Hokuyo UTM-30LX scanning laser rangefinder [28], two Stalker stationary speed sensors [29] and an industrial fanless minicomputer. The main technical characteristics of the sensors are shown in Tables 3 and 4, and the main technical
characteristics of the computer are shown in Table 5. The Hokuyo UTM-30LX scanning laser rangefinder is a 2D and single layer sensor.

Table 3. Main technical specifications of the Hokuyo UTM-30LX scanning laser range finder [28].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light source</td>
<td>Semiconductor laser diode (λ = 905 nm)</td>
</tr>
<tr>
<td>Detection range</td>
<td>0.1 to 30 m</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.1 to 10 m: ±30 mm, 10 to 30 m: ±50 mm</td>
</tr>
<tr>
<td>Scan angle</td>
<td>270°</td>
</tr>
<tr>
<td>Angular resolution</td>
<td>0.25° (360°/1440 steps)</td>
</tr>
<tr>
<td>Scan time</td>
<td>25 ms/scan</td>
</tr>
<tr>
<td>Interface</td>
<td>USB2.0 (Full Speed)</td>
</tr>
<tr>
<td>Command system</td>
<td>Exclusively designed command SCIP Ver.2.0</td>
</tr>
</tbody>
</table>

Table 4. Main technical specifications of the Stalker stationary speed sensor [29].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Stationary Doppler radar speed sensor</td>
</tr>
<tr>
<td>Communication interface</td>
<td>RS-232</td>
</tr>
<tr>
<td>Speed range</td>
<td>1 MPH to 200 MPH (1.6 to 321 km/h)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>+/−3%</td>
</tr>
</tbody>
</table>

Table 5. Main technical specifications of the computer.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Ubuntu 18.04.6 LTS</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>16 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>512 GB SSD</td>
</tr>
</tbody>
</table>

This configuration makes it possible to monitor two adjacent lanes at a toll, as illustrated in Figure 2. Each lane is equipped with a speed radar, and given the range of the laser sensor, a single device is used to monitor both lanes simultaneously.

The sensors were installed in a toll station, as shown in Figures 2–4. This station has 5 lanes, and data recording was carried out in the center lane and an adjacent one. The separators between lanes have a width of 2 m, while each lane has a width of 3.5 m.

Figure 3 presents a picture of the toll with the sensors installed. In it, the lanes designated as right and left are identified. The laser is installed in a structure at 1 m high, visible on the left side of the picture, while on the right side, we can see the structure that holds the speed sensors and the computer. The laser rangefinder was installed in the middle of two lanes with its light beams pointing toward the side of the vehicles to create 3D models of the side faces. Considering that the angular range of the sensor is 270°, it was installed with the connection cables pointing perpendicularly toward the ground as seen in Figure 4. The speed sensors were installed at a distance of 5 m from the laser rangefinder, with their focus pointing towards the center of the lane at the point through which the light beams of the laser rangefinder pass, as seen in Figure 2.
Figure 2. Top view of the sensor installation diagram: (1) Scanning laser rangefinder. (2) Speed sensors.

Figure 3. Sensors installed at the toll station: (a) Scanning laser rangefinder. (b,c) Speed sensors for the right and left lanes, respectively.

Figure 4. Front view of the laser rangefinder installation diagram.

Since the laser rangefinder has an angular range of 270°, a polar coordinate plane was defined as a reference frame for the scanner reading. In this plane, the 0° angle is
oriented towards the right lane in a horizontal direction parallel to the ground, as depicted in Figure 5. Consequently, the sensor makes measurements within an angular range that goes from $-45^\circ$ to $225^\circ$, according to the established polar coordinate plane. Figure 5 provides a visual representation of the angular range of the scanner with a front view from the position of the speed sensors.

![Laser rangefinder angular range](image)

**Figure 5.** Laser rangefinder angular range.

To ensure accurate data acquisition using the sensors, the following installation and operating conditions were prepared and strictly adhered to during the data collection period:

- The sensors must be protected from rain and direct sunlight, a protection that the toll station cover adequately provides.
- It is recommended that vehicles drive between 0.5 m and 2.5 m from the laser rangefinder to ensure enough point resolution. Given that the width of the lane separator is 2 m, it is established that the minimum distance at which vehicles travel is 1 m.
- There must be no obstructions between the vehicle and the ground to the laser rangefinder. The sensor has been installed without obstacles that prevent proper scanning of vehicles and the ground surface.
- Vehicles must travel at a maximum speed of 40 km/h to ensure adequate point density (in case of different circumstances, the possibility of changing the range sensor for one with a higher sampling rate should be considered). The lanes are equipped with speed bumps that limit the speed of vehicles to this maximum.
- Vehicles are required to drive in front of the laser rangefinder without stopping until the scan is completed. Since the sensor is located after the toll booth, vehicles do not normally stop in front of the sensor during normal toll operation.

### 2.2. Data Format

Vehicles are classified into six different categories, which generally represent the classes of vehicles used for classification at Colombian tolls. Table 6 presents a detailed breakdown of the vehicles obtained, including the classes they have been classified into, a brief description of each class, the number of vehicles per class, and visual examples of each.

**Table 6.** Vehicle types.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Quantity</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cars, campers and vans, with or without a trailer, ambulances, small 2-axle trucks with a single tire on the rear axle</td>
<td>31,432</td>
<td>![Example Image]</td>
</tr>
</tbody>
</table>
Table 6. Cont.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Quantity</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Minibuses with single tire on the rear axle</td>
<td>452</td>
<td><img src="image1.png" alt="Minibus" /></td>
</tr>
<tr>
<td>3</td>
<td>2-axle buses with double tires on the rear axle</td>
<td>1158</td>
<td><img src="image2.png" alt="Bus" /></td>
</tr>
<tr>
<td>4</td>
<td>Buses and trucks with 3 or more axles</td>
<td>1008</td>
<td><img src="image3.png" alt="Truck" /></td>
</tr>
<tr>
<td>5</td>
<td>Small 2-axle trucks with double tires on the rear axle</td>
<td>1179</td>
<td><img src="image4.png" alt="Small Truck" /></td>
</tr>
<tr>
<td>6</td>
<td>Large 2-axle trucks with double tires on the rear axle</td>
<td>797</td>
<td><img src="image5.png" alt="Large Truck" /></td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>36,026</strong></td>
<td></td>
</tr>
</tbody>
</table>

1 Trucks whose gross vehicle weight must be less than 8.5 tons. For example: light trucks, food transport trucks, cranes, etc. 2 Trucks whose gross vehicle weight must be greater than 8.5 tons, for example, heavy trucks, trucks for transporting materials, dump trucks, etc.

The dataset is composed of 36,026 object classes stored in text files with .out extension and organized into daily and hourly folders according to the date and time of capture. Figure 6 shows the storage structure of the dataset.

![Dataset storage structure](image6.png)

During the data acquisition process, the software automatically divides the laser light beams into two parts: one for the right lane and one for the left lane. This ensures that there are no problems if two vehicles pass simultaneously. Therefore, each file of the dataset corresponds to a single vehicle that passed through the toll station.
There are 12 folders at the root of the dataset, one for each day, and they are named with the corresponding year, month and day, for example, folder 20230401 contains the data for 1 April 2023. Each day’s folder can contain up to 24 subfolders, one for each hour of the day, which are named with the corresponding hour in two-digit format from 00 to 24. The files are named with the date and time of the capture, the lane by which the vehicle passed (D for the right lane and I for the left lane) and the type of vehicle (see Table 6). The following format is used for the file name:

```
year_month_day_hour_minute_second_lane_type.out
```

The scanner’s acquisition rate is 40 SPS (Scans Per Second), which means that the sensor performs a full sweep of distance measurements from $-45^\circ$ to $225^\circ$ every 0.25 ms. The angular resolution of the scanner is 0.25°, which implies that each scan or sample of the scanner contains 1081 values. Of these values, the first 541 are assigned to the right lane and the remaining 540 to the left lane.

Each file in the dataset is made up of a matrix, where each row represents a sample of the scanner. Figure 7 shows the general structure of the rows in each file, the upper array corresponds to the files of vehicles in the right lane and the lower one to the vehicles in the left lane.

![Figure 7. Data arrays for the right and left lanes, respectively.](image)

The number of rows in each file depends on the size of the vehicle, as shown in Figure 8. Each row is a set of float values separated by whitespace. The first value corresponds to the speed in m/s, measured with the speed sensor of the corresponding lane at the same time of the scan, and is highlighted in red in Figure 8. The second value indicates the relative time in seconds, obtained with a timer that starts with data acquisition, and is represented in blue. The remaining values are measurements from the laser rangefinder scan or sample and are identified in green. Therefore, for vehicles traveling in the right lane, the files will contain rows with 543 values, while for vehicles traveling in the left lane, the files will have 542 elements.

![Figure 8. Example of file content.](image)

For the 36,026 vehicles, the average speed measured was 14.2 km/h with a standard deviation of 3.7 km/h.
The data set is shared in its raw form, that is, comprising points with polar coordinates, allowing interested researchers to explore a wide range of analysis and processing options, including filtering, interpolation, analysis of sensor response, among others. As detailed in the following sections and following the instructions in Section 3.2.1 and Algorithm 3, a simple script can be created to convert the polar coordinates of the data set files into a Cartesian coordinate matrix. This matrix can be saved in a PCD file, which is compatible with software such as CloudCompare [30] or other similar programs.

3. Methods

This section addresses the software tools used to read and interpret files, allowing the raw point clouds to be viewed. Likewise, the processing algorithms used to generate point clouds with the lateral profiles of the vehicles are detailed, eliminating background information, the ground surface and noise. This provides a clear display of the vehicle type and its distinctive features. In addition, the methodology and tools used for validation and labeling of the dataset are presented.

The data acquisition software was developed in Python 3.69. The interface with the laser rangefinder was made with the HokuyoAIST library [31,32] and the Point Cloud Library (PCL) [33] version 1.10.1 was used to process the point clouds.

3.1. Data Acquisition

Algorithm 1 presents a brief description of the computational steps invoked to acquire the range and speed data of the vehicles, as well as their storage in the files that constitute the dataset. In summary, the algorithm is responsible for the initial configuration, activation and reading of the sensors, the selection of valid samples to determine their belonging to a vehicle, and finally the registration of the valid samples in the corresponding file.

Algorithm 1 The main computational steps to acquire vehicle range data.

1: right_finalized ← False // indicates when capture of a vehicle in right lane ends
2: left_finalized ← False // indicates when capture of a vehicle in left lane ends
3: initial_time ← read current time // read the current time
4: turn on sensors // turn on laser rangefinder and speed sensors
5: while True do
6: scan ← read lidar // read a LiDAR sample
7: right_speed ← read right sensor // read a value from right lane speed sensor
8: left_speed ← read left sensor // read a value from left lane speed sensor
9: if right_speed > 0 then // vehicle detection in right lane if speed > 0
10: time ← read current time − initial_time // estimate relative time
11: right_array ← {right_speed time scan[0..540]} // create array of the sample
12: // detects a vehicle and selects valid samples
13: right_laser_block, right_finalized ← SAMPLESSELECTION(right_array)
14: if right_finalized is True then
15: // save data to file if vehicle detection ends
16: save right_laser_block to file
17: right_finalized ← False
18: end if
19: end if
20: if left_speed > 0 then // vehicle detection in left lane if speed > 0
21: time ← read current time − initial_time
22: left_array ← {left_speed time scan[541..1080]} // read a value from left lane speed sensor
23: left_laser_block, left_finalized ← SAMPLESSELECTION(left_array)
24: if left_finalized is True then
25: save left_laser_block to file; left_finalized ← False
26: end if
27: end if
28: end while
The proper selection of valid samples is essential to determine the precise moment at which a vehicle passes in front of the laser sensor, thus marking the beginning of the creation of a new laser block. This block records all light beams captured during vehicle detection, except those associated with a speed equal to zero. To carry out this process, a region of interest (ROI) is established whose dimensions are adjusted according to the specific location, being activated or deactivated depending on the presence or absence of an object.

Since the laser sensor provides the range measurements in polar format, the region of interest is configured in a cone shape, with parameters such as a minimum radius (minimum distance), a maximum radius (maximum distance), and top and bottom opening angles. These parameters are centered at the point (0, 0) corresponding to the sensor position and are adjusted independently for each lane under system supervision. Figure 9 illustrates the region of interest (represented by red cones) when selecting valid samples. For the right lane, a region of interest has been defined that spans from $-30^\circ$ to $30^\circ$ in angle, and from 1 m to 3 m in distance. As for the left lane, the region of interest extends from $150^\circ$ to $210^\circ$ in angle, maintaining the same distance range, that is, from 1 m to 3 m.

![Region of interest for vehicle detection.](image)

To guarantee good performance of the algorithm, it is essential to ensure that the region of interest does not include light beams that impact the ground or surrounding surfaces, such as walls or objects in the environment, as this could cause erroneous activations in the region of interest.

In Algorithm 1, it begins with the acquisition of data by the laser and speed sensors. Subsequently, a validation is carried out to discard samples in which the recorded speed is equal to 0. Those samples with speeds other than 0 are subjected to the process of selecting valid samples. On the other hand, Algorithm 2 offers a synthesis of the computational steps necessary for this selection. Once the region of interest (ROI) has been defined by adjusting the parameters, and in the absence of vehicle detection, the activity in that region is monitored. Detection of activity in the ROI indicates the creation of a new laser block. Once a vehicle is detected by activating the ROI, its continued presence is verified. Otherwise, a countdown is started to stop the block and send it to the storage stage.
Algorithm 2 The main computational steps for the selection of valid samples.

1: ROI // set region of interest
2: count ← 0 // counter continuous samples that are outside ROI
3: status ← waiting // algorithm status: waiting or collecting a vehicle
4: declare buffer[10, 543] // temporary buffer that stores last 10 samples
5: laser_block ← [] // laser block with all samples of the vehicle
6: function SAMPLES_SELECTION(array)
7: if buffer is full then
8: rotate buffer // rotates buffer and remove oldest sample
9: append array to buffer // append the sample to buffer end
10: if status is waiting then
11: laser_block ← [] // cleans laser block
12: // If 10% of sample points belong to ROI, vehicle data recording begins
13: if 10% of array ∈ ROI then status ← collecting
14: end if
15: else
16: append buffer[0][all] to laser_block // append the buffer oldest sample to laser block
17: // counts continuous samples with 90% of points outside ROI (deactivation) to detect object
18: // completion
19: if 90% of array ∈ ROI then count ← count + 1
20: else count ← 0
21: end if
22: if count > 20 then
23: // if more than 20 continuous samples have 90% of points outside ROI, object detection ends
24: status ← waiting; count ← 0
25: // if the laser block has more than 30 samples, it is considered a valid vehicle
26: if size of laser_block > 30 then return laser_block, True
27: end if
28: end if
29: else append array to buffer
30: end if
31: return laser_block, False
32: end function

3.2. Data Processing

The data files must be processed to obtain a point cloud of the vehicle, in which
the stage background, the ground surface and the associated noise have been removed.
A general diagram of this process can be seen in Figure 10. The following subsections show
each of the four stages presented in the diagram in more detail.

3.2.1. Creation of the Point Cloud

In the initial phase, the process involves reading the vehicle’s range data from a
file, to generate a three-dimensional representation of the captured scene. For this representa-
tion, a Cartesian coordinate system is established where the Z axis denotes the
height of the vehicle, the X axis the length, and the Y axis the width. Each row of the file
contains scan data expressed in polar coordinates (r, θ), where r indicates the distance
measured in mm and θ represents the corresponding angle. For the right lane, the angles
vary from −45° to 90° in 0.25° increments, while for the left lane they range from 90.25°
to 225°. The construction of the point cloud requires the conversion of the range data to
Cartesian coordinates (Y, Z) and the adaptation of the measurement units to m, using
Equations (1) and (2).
\[ Y = r \cos \left( \frac{\pi}{180} \theta \right) / 1000 \]  
\[ Z = r \sin \left( \frac{\pi}{180} \theta \right) / 1000 \]  

The X-axis coordinate is obtained using Equation (3), where \( X_{\text{prev}} \) is the X coordinate of the previous sample (this variable is initialized to 0), \( \text{speed} \) is the measured speed of the sample in m/s and is the first value in the array of each row in the range data file, \( \text{time}_{\text{curr}} \) is the relative time in seconds of the sample and is the second value in the array, and \( \text{time}_{\text{prev}} \) is the relative time in seconds of the previous sample, that is, in the array of the row immediately preceding the sample being processed.

\[ X = X_{\text{prev}} + \text{speed} \cdot (\text{time}_{\text{curr}} - \text{time}_{\text{prev}}) \]  

Algorithm 3 presents a brief description of the computational steps used to construct the point cloud from the dataset files and Figure 11 shows the 3D image of a point cloud built from the data in one of the files. It is possible to observe the entire captured scene, including the vehicle, the ground surface and the background.

![Figure 11. Three-dimensional image of a raw point cloud.](image)

**Algorithm 3** The main computational steps for the creation of the point cloud.

```plaintext
1: data ← read file
2: angular_resolution ← 0.25 // angular resolution of LiDAR
3: angular_range ← 270 // angular range of LiDAR
4: if lane is right then angle_array ← \{-45 : 0.25 : 90\} // angle array of right lane
5: else angle_array ← \{90.25 : 0.25 : 225\} // angle array of left lane
6: end if
7: speed_data ← data[all][0] // speed array of all samples
8: time_data ← data[all][1] // time array of all samples
9: samples_number ← rows of data // number of samples or scans
10: points_number ← columns of data − 2 // number of values or points per sample
11: \( X_{\text{prev}} \) ← 0 // first position of the X axis
12: X, Y, Z // point cloud coordinate array
13: for \( i \) ← 1, samples_number do
14: // calculates the point cloud coordinates for each sample
15: speed ← speed_data[\( i \)]
16: append \{\( X_{\text{prev}} + \text{speed} \cdot (\text{time_data}[\( i \]) - \text{time_data}[\( i - 1 \]) \} \} \text{ones(points_number)}\} to X
17: \( X_{\text{prev}} \) ← X[\( end \]
18: append \{data[\( i \)]\} ends \( \times \text{cos(angle_array} \times \pi / 180) / 10000\} to Y
19: append \{data[\( i \)]\} ends \( \times \text{sin(angle_array} \times \pi / 180) / 10000\} to Z
20: end for
```
3.2.2. Distance Filtering

Once the point cloud is created, the next step is to eliminate non-relevant information, such as pedestrians walking in front of the sensor, background elements, other sensors at the toll, and vehicles traveling in other lanes. All information from these unnecessary areas is removed by distance filtering, that is, thresholding in terms of height (Z axis) and depth (Y axis) in the point cloud.

The distance filtering limits were set between 1 m and 3.5 m in the Y axis and between −1.5 m and 3 m in the Z axis, considering that the origin is the position of the laser rangefinder, as seen in Figure 12. To define these limits, the width of the lane and the maximum height of the vehicles were considered.

![Figure 12. Region for distance filter.](image)

Figure 13 shows the point cloud of Figure 11 after applying distance filtering. The 3D representation of the filtered point cloud is shown on the left, while on the right, it is presented in 2D format. The elimination of background elements can be seen, leaving only the information about the vehicle and the surface of the lane through which it travels.

![Figure 13. Point cloud with distance filtering.](image)

3.2.3. Ground Surface Extraction

One of the most common processes in the segmentation of elements in a point cloud is the estimation of parametric shape models (planes, cylinders, circles, etc.), facilitating the detection of elements in the scene. To identify and manipulate the ground surface present
in the point cloud, a plane extraction algorithm was used. The identification of the planar model of the ground surface allows for the elimination of information present on the road surface. For the segmentation of the ground model, the implementation of the techniques present in Refs. [34,35] was used, which uses the RANSAC (RANdom SAmple Consensus) estimation method.

The method is used to estimate the model of a dominant plane perpendicular to the Z axis, that is parallel to the ground. The plane model has the form of Equation (4). First, the model of a plane is identified with the greatest number of points that fit it, which must be at a distance from the plane less than a defined distance threshold of 5 cm and with a normal approximately parallel to that of the plane, that is, with an angle between them less than an angle threshold defined by 10°. Subsequently, the points that fit the found plane are eliminated from the point cloud, which corresponds to the ground surface. Algorithm 4 outlines the basic steps that are used to extract the ground surface.

\[ ax + by + cz + d = 0 \]  

**Algorithm 4** The main computational steps for ground surface extraction.

1. \( \text{iterations} \leftarrow 5000 \)
2. \( d \leftarrow 0.05 \) // set distance threshold
3. \( \epsilon \leftarrow 10 \) // set angle epsilon threshold
4. \( P \leftarrow \{ p_1 \ldots p_n \} \) // set of 3D points
5. \( P_\perp \leftarrow \{ p_i | \mathbf{n}_i \times \mathbf{z} \approx 0 \} \) // select set \( P_\perp \) with normals parallel to \( \mathbf{z} \), angle between \( \mathbf{n}_i \) and \( \mathbf{z} \) \( \leq \epsilon \)
6. for \( i \leftarrow 1 \), \text{iterations} do
7. \( C_i \leftarrow \text{random} \ P_\perp \subset P_\perp \) // select a random subset of \( P_\perp \)
8. // find the best plane fit using sample consensus
9. \( \text{estimate} \{ a, b, c, d \}, A = a \cdot p_j^x + b \cdot p_j^y + c \cdot p_j^z + d, p_j \in C_i \)
10. \( P_i \leftarrow \{ p_j \in P_\perp | p_j \text{ fits to } A \} \) // select the set that fit to \( A \), distance between \( p_j \) and \( A \) plane \( \leq d \)
11. end for
12. \( P^* \leftarrow P - \text{larger}(P_i) \) // point cloud without ground surface, ground surface is set with the most points

Figure 14 shows a segmented point cloud, depicting the vehicle information in black and the ground surface in red. The precise identification of the ground surface stands out, which can be successfully eliminated.

![Figure 14. Point cloud with ground surface extraction.](image)

### 3.2.4. Statistical Filtering

Finally, statistical filtering is used to remove noise from the scene by implementing algorithms described in Refs. [34,36]. Noise appears for several reasons such as dust in the environment, uncertainty in the measurement, etc. In point clouds, this noise can be observed as isolated points without any relationship with the real scene.

The method calculates the mean distance \( \overline{d} \) of each point to its \( k \) closest neighbors. It then estimates the mean \( \mu \) and standard deviation \( \sigma \) of the mean distance space of all points. Points whose mean distance \( \overline{d} \) to their \( k \) closest neighbors is approximate to
the mean $\mu$ in a range no greater than $\alpha$ times the standard deviation $\sigma$ are retained. That is, the remaining filtered point cloud $P^*$ can be estimated by Equation (5) where $P$ is the complete point cloud. The parameter $k$ was set to 50 closest neighbors and the factor $\alpha$ to 2.5. The computational steps of the statistical filtering method are presented in Algorithm 5.

$$P^* \leftarrow \{ p^*_i \in P | (\mu - \alpha \cdot \sigma) \leq d^*_i \leq (\mu + \alpha \cdot \sigma) \}$$  \hspace{1cm} (5)

**Algorithm 5** The main computational steps for statistical filtering.

1. $k \leftarrow 50$
2. $\alpha \leftarrow 2.5$
3. $P \leftarrow \{ p_1 \ldots p_n \}$ \hspace{1cm} // set of 3D points
4. for all $p_j \in P$ do
5. estimate $\mathcal{P}_k$ \hspace{1cm} // set of $k$ closest neighbors to $p_j$
6. for all $p_j \in \mathcal{P}_k$ do
7. $d_{ij} \leftarrow \{|p_i - p_j|\}$ \hspace{1cm} // compute the distances between $p_i$ and its $k$ closest neighbors
8. end for
9. $\bar{d}_i \leftarrow \text{mean}(d_{ij} \text{ for } j \leftarrow [1 \ldots k])$ \hspace{1cm} // compute mean distance of $p_i$ to its $k$ closest neighbors
10. end for
11. $\mu \leftarrow \text{mean}(\bar{d}_i \text{ for } i \leftarrow [1 \ldots n])$ \hspace{1cm} // compute mean of mean distance space of all points
12. $\sigma \leftarrow \text{stddev}(\bar{d}_i \text{ for } i \leftarrow [1 \ldots n])$ \hspace{1cm} // compute standard deviation of mean distance space of all points
13. $P^* \leftarrow \{ p^*_i \in P | (\mu - \alpha \cdot \sigma) \leq d^*_i \leq (\mu + \alpha \cdot \sigma) \}$ \hspace{1cm} // filtered point cloud

On the left in Figure 15, the filtered point cloud is shown in black, and the removed points are in red. On the right, only the filtered point cloud is shown, where greater homogeneity can be observed in the surfaces and how some scattered points disappear.

![Figure 15. Point cloud with statistical filtering.](image)

### 3.3. Dataset Validation and Labeling

Once the range data has been obtained and stored in files, it is essential to perform two fundamental steps. First, extensive validation must be performed to ensure the accuracy and reliability of the data. Secondly, it is necessary to label each file with the corresponding vehicle type, which will allow its use in training automatic classification models. In this specific project, this process was carried out visually and manually, with the support of videos from cameras at the toll.

To streamline and systematize this process, a Graphical User Interface (GUI) was developed in Python. This tool makes it easy to view each point cloud individually, giving the user the ability to verify and validate the accuracy, integrity, and consistency of the data. Additionally, it allows the selection of the vehicle type from a predefined list of options. Once the vehicle type is selected, the tool automatically renames the file, adding an identification number corresponding to that vehicle type.

The creation of this GUI considerably simplified the validation and labeling process, while ensuring greater accuracy in assigning vehicle categories and validating the quality of the generated dataset. Figure 16 shows the interface developed for this purpose, providing a clear and friendly view of the labeling workflow.

![Figure 16. Interface developed for dataset validation and labeling.](image)
4. Application Examples

The purpose of this section is to offer a broader view of the possible applications of the dataset, as well as to explore some potential usage scenarios. In particular, we will focus on two key areas: automatic classification and 3D modeling.

4.1. Application for Automatic Classification

Vehicle classification represents a fundamental field of study, given its relevance in a wide range of applications ranging from surveillance and security systems to traffic management and accident prevention, among others. In this context, a variety of algorithms aimed at accurate object classification have been developed and applied, many of which are based on the processing of point clouds obtained by LiDAR sensors.

Some of the techniques for point cloud classification include: neural networks [37–39], Bayesian networks [40], K-nearest neighbors KNN [41], support vector machines SVM [42,43], among others.

In this work, to demonstrate that the dataset is well constructed and labeled, a classifier was implemented and trained from a support vector machine (SVM). The input data to the classifier is geometric information of the vehicles. Figure 17 shows the stages of the feature extraction and classification process.

According to the vehicle types in Table 6, the classification depends largely on the number of tires touching the ground and the dimensions of the vehicles. For this reason, a method was implemented to extract geometric characteristics of vehicles from the information contained in the point cloud. In this work, the following five geometric features were used: number of tires touching the ground, length and height of the vehicle, diameter of the first tire, and height of the vehicle on the first axle (see Figure 18).
Finally, a classifier based on an SVM with a polynomial kernel was trained. The dataset is used with a distribution of 80% point clouds for training and 20% for validation. To balance the number of samples of each vehicle type, only 3% of the point clouds from class 1 (cars, campers, vans) were used. The training results are shown in Figure 19 using a confusion matrix.

In general, a very good performance of the classifier is observed. For the “Cars” and “+3-axles Trucks” classes, the performance is above 95%. With the “Minibuses” class a performance of 79% is achieved and some samples are confused with the “Cars” class, this may be due to some minibuses being as small as vehicles. Similarly, the “Buses” class achieves a performance of 84% and some samples are confused with minibuses for similar reasons to the previous case. In the case of the “2-axle Truck” class, although good performance is also achieved, in the labeling process it is sometimes difficult to differentiate between a large truck and a small one, which according to the regulations has to do with the size of the tires.

It is crucial to highlight that the SVM classifier presented in this study is used only for demonstration purposes on the data set. However, employing a more advanced classifier could lead to higher classification accuracy.

4.2. Application for 3D Modeling

The three-dimensional creation of objects and scenarios is a constant challenge and of great interest in the communities dedicated to graphics, computer vision and photogrammetry. Three-dimensional digital models are essential in a variety of applications, including inspection, navigation, object recognition, visualization and animation [44].

In this study, due to the characteristics and installation method of the data acquisition equipment, only one face of each vehicle is captured in the point cloud dataset. To fill in the missing coordinate points in the unscanned planes, 3D surface reconstruction techniques are employed, as described in Refs. [45–48].

In the experiment, the scanned face was mirrored and separated from the original by a distance approximately the width of the vehicle. Point clouds were then added to the front, back, top and bottom faces, followed by a plane reconstruction using the Poisson method with the CloudCompare tool [30].

Figure 20 exemplifies the result of 3D modeling of the point cloud dataset. Details such as the side-view mirror and the fender can be seen, demonstrating the wealth of detailed vehicle information in this dataset and its usefulness in 3D modeling.
5. Conclusions

This work addresses the creation of a point cloud data set obtained by laterally scanning vehicles at a toll station, using a laser rangefinder and two Doppler speed sensors. Several range image processing stages, such as distance filtering, ground surface removal, and statistical filtering, were applied to obtain vehicle side profiles without noise and irrelevant information.

The manual labeling and validation process used to classify vehicles into six distinct classes is described; although, the structure of the dataset allows for flexibility in relabeling vehicles based on the specific application.

In addition, two examples of applications of the dataset were presented. In the first, a classification method based on geometric features was proposed using a support vector machine. In the second, a 3D modeling application was demonstrated using three-dimensional reconstruction techniques.

Notwithstanding the above, this dataset provides researchers and developers with the opportunity to test and validate range image processing and classification methods and algorithms, such as deep learning and support vector machines, among others.

The main contribution of this dataset is the 36,026 point clouds of side views of a good variety of vehicles. Therefore, the dataset is an important contribution to research in computer vision and computational intelligence.

In future work, the authors promise to acquire more data and label it to expand the data set. In addition, efforts will be devoted to modifying and testing the implemented processing and classification methods in order to improve the classification accuracy. It is also planned to integrate video cameras into the system for license plate detection and thus add more information to the data set.


**Funding:** This work is part of a beneficiary project from the National Doctorate in Business call number 758-2016, funded by the Ministry of Science, Technology and Innovation of Colombia, the Universidad del Valle, and the company SSI Soluciones y Suministros para Ingenierías SAS.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are openly available at https://doi.org/10.5281/zenodo.10974361.

**Acknowledgments:** We thank the Autopistas del Café concession for allowing us access to one of their toll stations to acquire the data presented in this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.
Abbreviations
The following abbreviations are used in this manuscript:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITS</td>
<td>Intelligent transportation systems</td>
</tr>
<tr>
<td>LiDAR</td>
<td>Light detection and ranging</td>
</tr>
<tr>
<td>PCL</td>
<td>Point Cloud Library</td>
</tr>
<tr>
<td>RANSAC</td>
<td>RANdomized SAmple Consensus</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of interest</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
</tbody>
</table>

References


27. Li, Y.; Tok, A.Y.; Sun, Z.; Ritchie, S.G.; Altu, K.R. LiDAR Vehicle Point Cloud Reconstruction Framework for Axle-Based Classification. IEEE Sens. J. 2023, 23, 11168–11180. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.