

# Investigation of Truck Data Collection using LiDAR Sensing Technology along Rural Highways

September  
2021

A Research Report from the Pacific Southwest  
Region University Transportation Center

Stephen G. Ritchie, University of California, Irvine

Andre Tok, University of California, Irvine

Yiqiao Li, University of California, Irvine

Jared Sun, University of California, Irvine

Koti R. Allu, University of California, Irvine



[page intentionally left blank]

## TABLE OF CONTENTS

### Contents

About the Pacific Southwest Region University Transportation Center .....	5
U.S. Department of Transportation (USDOT) Disclaimer .....	6
California Department of Transportation (CALTRANS) Disclaimer .....	6
Disclosure.....	7
Acknowledgments.....	8
Abstract.....	9
Executive Summary.....	10
1. Introduction.....	17
2. Literature Review .....	18
3. Data Description and Preprocessing .....	22
3.1 Study Site Layout.....	22
3.2 Data Collection Setup.....	22
3.3 Data Description.....	24
3.4 Semi-automatic Data Labeling Method .....	25
3.5 Data Preprocessing.....	27
4. Vehicle Point Cloud Registration Framework.....	31
4.1 Introduction to Point-set Registration .....	31
4.2 Probabilistic Point-set Registration.....	32
4.3 Vehicle Point Cloud Registration Framework .....	33
5. FHWA Axle-based Classification .....	42
5.1 Feature Extraction.....	42
5.2 Bootstrap Aggregating Deep Neural Network for Vehicle Classification.....	44
5.3 Model Results.....	45
6. Truck Body Type Classification .....	49
6.1 PointNet-based Truck Classification Model .....	49
6.2 Model Averaging .....	52
6.3 Model Results.....	53
7. LiDAR intensity-based Truck Surface Characterization .....	60
7.1 Introduction.....	60

7.2	Proposed LiDAR Intensity Homogenization Framework.....	61
7.3	Qualitative Results for LiDAR Intensity Homogenization.....	63
7.4	Future Expansion.....	64
8.	Conclusion .....	66
	Reference .....	67

## About the Pacific Southwest Region University Transportation Center

The Pacific Southwest Region University Transportation Center (UTC) is the Region 9 University Transportation Center funded under the US Department of Transportation's University Transportation Centers Program. Established in 2016, the Pacific Southwest Region UTC (PSR) is led by the University of Southern California and includes seven partners: Long Beach State University; University of California, Davis; University of California, Irvine; University of California, Los Angeles; University of Hawaii; Northern Arizona University; Pima Community College.

The Pacific Southwest Region UTC conducts an integrated, multidisciplinary program of research, education and technology transfer aimed at *improving the mobility of people and goods throughout the region*. Our program is organized around four themes: 1) technology to address transportation problems and improve mobility; 2) improving mobility for vulnerable populations; 3) Improving resilience and protecting the environment; and 4) managing mobility in high growth areas.

## **U.S. Department of Transportation (USDOT) Disclaimer**

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

## **California Department of Transportation (CALTRANS) Disclaimer**

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the United States Department of Transportation's University Transportation Centers program, in the interest of information exchange. The U.S. Government and the State of California assume no liability for the contents or use thereof. Nor does the content necessarily reflect the official views or policies of the U.S. Government and the State of California. This report does not constitute a standard, specification, or regulation. This report does not constitute an endorsement by the California Department of Transportation (Caltrans) of any product described herein.

## Disclosure

Principal Investigator, Co-Principal Investigators, others, conducted this research titled, “Investigation of Truck Data Collection using LiDAR sensing technology along Rural Highways” at the Institute of Transportation Studies, University of California, Irvine. The research took place from 05/01/2020 to 09/30/2021 and was funded by a grant from the State of California, Department of Transportation (Caltrans) in the amount of \$130,000. The research was conducted as part of the Pacific Southwest Region University Transportation Center research program.

## Acknowledgments

This study was made possible through funding received by the California Department of Transportation (Caltrans) and the Pacific Southwest Region University Transportation Center (PSR-UTC). The authors would like to thank Caltrans and PSR-UTC for their support of university-based research, and especially for the funding received for this project. The authors also gratefully appreciate the support provided by Tom Shepard of Caltrans District 11 for his numerous efforts in providing access to the study site and assistance in setting up field sensors. The contents of this research reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This research does not constitute a standard, specification, or regulation.

## Abstract

Heavy trucks comprise a much larger proportion of overall traffic in rural highways compared with their urban counterparts, hence detailed classification counts are needed to adequately assess the impacts of truck activity in these regions. Under current practice, truck count data obtained along non-detectorized rural highway corridors are either estimated using unreliable growth factors applied on decades-old observed data or collected via pneumatic tubes which need to be laid across highways to collect traffic data. This study investigated a new truck classification approach using a Light Detection and Ranging (LiDAR) sensor array in a horizontal orientation, utilizing a reconstruction procedure that combines individual LiDAR frames with sparse point clouds to generate a feature-rich dense point cloud representation of vehicle objects to facilitate accurate truck classification. Two LiDAR-based classification models were developed in this study: an axle-based model following the FHWA-CA scheme and a detailed body classification model. The axle-based model demonstrated the ability to distinguish vehicle classes according to the FHWA-CA scheme on a truck-focused dataset with a correct classification rate (CCR) of 0.79, averaged across all classes. The corresponding CCR for the body classification model was 0.88 across 31 body classes. A preliminary investigation of LiDAR intensity on trailer surfaces was also performed to evaluate the potential of identifying fleet characteristics of trucks.

# Investigation of Truck Data Collection using LiDAR Sensing Technology along Rural Highways

## Executive Summary

Heavy trucks comprise a much larger proportion of overall traffic in rural highways compared with their urban counterparts, hence detailed classification counts are needed to adequately assess the impacts of truck activity in these regions. The traffic monitoring guide recommends that at least 30 percent of traffic data collection sites provide classification data. However, piezo-based sensors, which are the predominant technology for collecting truck classification data are costly to maintain.

Under current practice, truck count data obtained along non-detectorized rural highway corridors are either estimated using unreliable growth factors applied on decades-old observed data or collected via pneumatic tubes which need to be laid across highways to collect traffic data. The former approach is inaccurate and tends to underestimate the actual traffic growth in California. The latter approach exposes field personnel to significant safety hazards and is strongly discouraged unless lane closures are performed. Notwithstanding, reliable rural highway data is still required to obtain necessary federal funding allocation through the Highway Performance Measurement System (HPMS) to ensure adequate maintenance of the rural highway network and monitor the performance of freight activity in these regions.

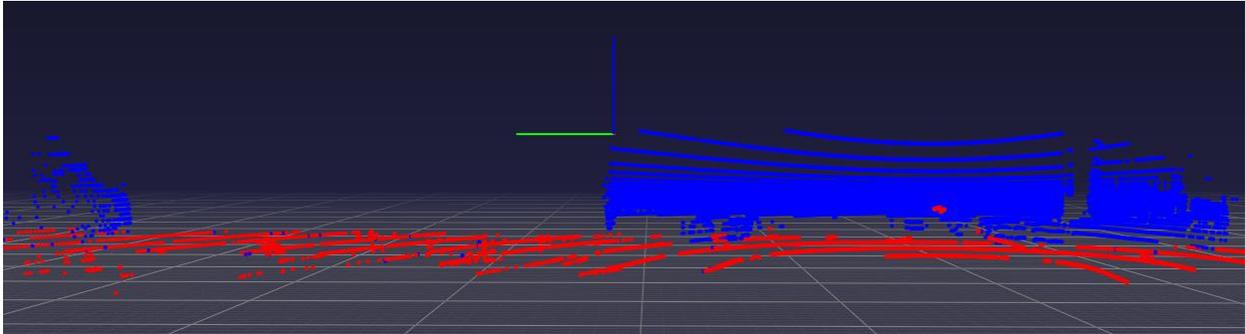
This study explored the use of Light Detection and Ranging (LiDAR) technology to develop new truck classification models, utilizing a reconstruction procedure that combines frames of sparse point clouds to generate a dense point cloud representation of vehicle objects to facilitate accurate truck classification, while preserving the panoramic LiDAR Detection Zone (LDZ). The lower profile of the reconstructed vehicle point clouds was extracted and used to classify vehicles based on the FHWA-CA scheme. Next, this study adopted the PointNet deep representation learning algorithm to train the classification model from the preprocessed point cloud data to classify trucks according to their detailed body configurations. A preliminary investigation of LiDAR intensity on trailer surfaces was also performed to evaluate the potential of identifying fleet characteristics of trucks.

### Data

The data used in this study were collected from the entrance ramp to the San Onofre truck scale from the Southbound I-5 Freeway in Southern California. The LiDAR sensor was placed in horizontal orientation above a traffic cabinet and was configured to scan the surroundings at a frequency of 10 rotations per second with a 180-degree LDZ – each rotation generating a single 3D point cloud frame.

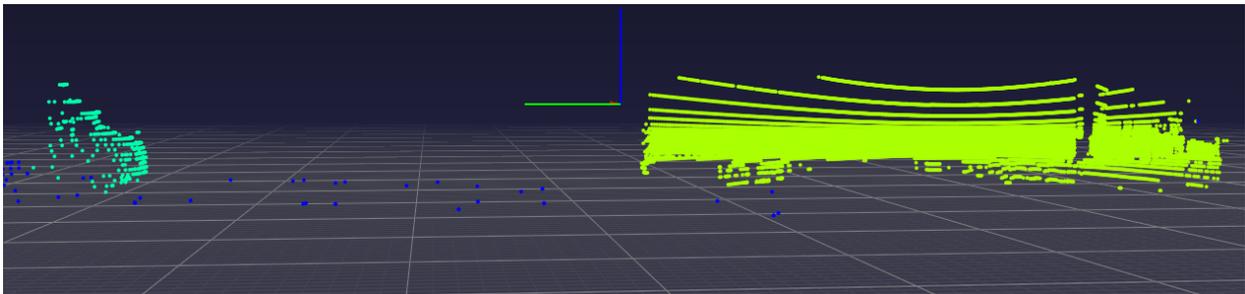
### Data Preprocessing

The first step in data preprocessing involved background subtraction – the removal of the large set of points in the raw point cloud not associated with vehicles as shown in Figure 1.



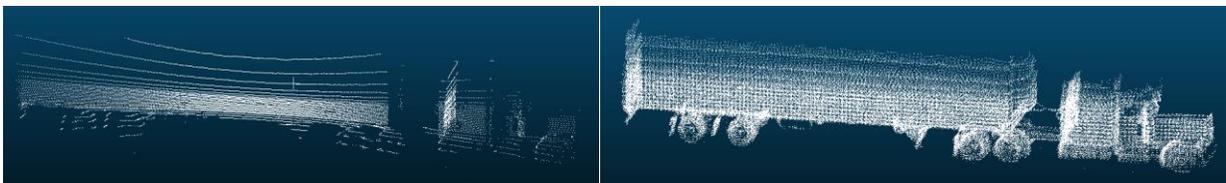
**Figure 1. Distinguishing foreground vehicle objects (blue) from the background (blue) to facilitate background subtraction**

This is followed by object detection, which determines the individual vehicles objects in the LiDAR field of view (shown in Figure 2) using a popular clustering technique called density-based spatial clustering of applications with noise (DBSCAN).



**Figure 2. Object Detection Result (Blue: outliers; Green: vehicle 1; Yellow: vehicle 2)**

The third step involves identifying the same vehicle point cloud object across multiple consecutive LiDAR frames through a process known as data association. These individual frames of the same vehicle are then combined to yield a dense reconstructed LiDAR representation of the vehicle.



**Figure 3. A truck object represented by a sparse point cloud from a single frame (Left) vs. a dense point cloud representation reconstructed from multiple frames (Right).**

Two classification models were developed from this preprocessed dataset: an FHWA axle-based classification model as well as a body configuration-based model.

### FHWA Axle-based Classification

For the axle-based model, essential features from the lower profile of the reconstructed truck point cloud were extracted and used as inputs for the vehicle classification model. Then, a deep ensemble neural network model was developed to assign vehicle point clouds to their corresponding FHWA-CA classes.

Table 1 provides a comparison between the model developed in this study with the state-of-the-art LiDAR-based classification model which used the single frame of an object to classify vehicles on the basis of the FHWA scheme (9).

**Table 1. Comparison between developed model (Bagging DNN) vs state of the art (Random Forest)**

FHWA-CA	CCR (Bagging DNN)	Testing Samples	Classes defined in (9)	CCR (Random Forest) (9)	Testing Samples (9)
Class 2	0.75	20	Passenger Vehicle	0.84	150
Class 3			Four-tire Single Unit	0.70	69
Class 4	None	None	Bus	1.00	20
Class 5 <sup>1</sup>	0.97	934	Two-axle, six-tire, single-unit truck	0.44	17
Class 6	0.95	208	Three-axle, single-unit truck	0.00	4
Class 7	0.76	17	Four or fewer axle, single-trailer truck	None	None
Class 8	0.84	117	None	None	None
Class 9 <sup>2</sup>	0.99	1,746	Five-axle, single-trailer truck	1.00	17
Class 10	0.33	12	None	None	None
Class 11	0.85	13	None	None	None
Class 12	0.50	2	None	None	None
Class 13	None	None	None	None	None
Class 14	1.00	31	None	None	None
<b>Average CCR</b>	0.79	-	-	0.76	-

Note: <sup>1</sup>Class 5 used in this study contained a two-axle truck pulling a small trailer which was not included in (9). <sup>2</sup>In the FHWA-CA scheme, Class 9 type 32 was separated from the rest of Class 9 truck and labeled as Class 14. In (9), Class 14 trucks are merged into Class 9 trucks.

Compared to the previous model (9), the new classification framework proposed in this study is able to classify vehicles in much more detail with significantly higher accuracy, especially for heavy-duty trucks from Class 8 to Class 14 which have disproportionately adverse impacts on pavement (39) and the environment (40).

### Truck Body Type Classification

Transportation agencies have been increasingly interested in collecting truck body configuration data due to its strong association with industries and freight commodities, to better understand their distinct operational characteristics and impacts on infrastructure and the environment. The body classification model was based on a novel deep neural network architecture called PointNet, which has the ability to directly adopt point clouds as inputs and detect critical features for classification from the raw inputs. An ensemble of five models were trained using different initialization values to address model variance

and further enhance model performance. Model averaging was applied to these five models to yield the final model prediction. Two model averaging methods were explored and in this study Simple Model Averaging (SMA) and Bayesian Model Averaging (BMA). The results from SMA and BMA are presented in Table 2. After applying model averaging across five PointNet models, the number of classes with CCR value less than 80 percent was significantly reduced. The model ensemble outperformed most of the individual models in terms of accuracy, average class CCR, and F1 score. SMA and BMA presented the same level of accuracy according to these aggregated measurements. The two-sided non-parametric Wilcoxon signed-rank test (47), was conducted to test if a significant difference existed between the results of SMA and BMA. The p-value of 0.02 showed the null hypothesis for the difference between SMA and BMA was significant, and at a significance level of 5 percent could be rejected. Therefore, the performance of SMA was determined to be significantly better than BMA.

**Table 2 Results of Body Classification Model**

	Model 1	Model 2	Model 3	Model 4	Model 5	SMA	BMA	Test Sample
20ft Container	0.98	0.98	1.00	1.00	0.93	1.00	1.00	59
40ft Container	1.00	1.00	1.00	1.00	1.00	0.99	0.99	196
53ft Container	0.94	0.97	0.97	0.97	0.97	0.96	0.96	170
Auto (Conventional)	0.71	0.94	0.91	0.88	0.82	0.91	0.91	34
Auto (Pickup)	0.90	0.95	0.79	0.90	0.74	0.89	0.89	19
Bobtail	1.00	1.00	1.00	1.00	1.00	1.00	1.00	109
Cab-over Enclosed Van (SU)	0.95	0.98	0.98	0.99	0.97	0.99	1.00	148
Concrete	1.00	1.00	1.00	1.00	1.00	1.00	1.00	16
Conventional Enclosed Van (SU)	0.97	0.96	0.97	0.97	0.97	0.97	0.97	362
Drop Frame Van (Semi)	0.82	0.82	0.79	0.82	0.86	0.82	0.79	28
Dry Bulk Transport	1.00	1.00	0.94	0.94	0.94	1.00	1.00	16
Enclosed Van (Multi)	0.80	0.80	1.00	0.80	0.80	0.80	0.80	5
Enclosed Van (Semi)	0.99	0.98	0.99	0.98	0.99	0.98	0.99	928
End Dump (SU)	0.81	0.89	0.92	0.89	0.89	0.88	0.88	26
End Dump (Semi)	0.83	1.00	0.96	0.91	0.87	0.96	0.96	23
End Dump wTrailer	1.00	1.00	1.00	1.00	1.00	1.00	1.00	7
Low Boy Platform	0.82	0.84	0.89	0.82	0.93	0.91	0.91	56
Low Loading	0.86	0.91	0.88	0.90	0.90	0.90	0.89	116
Open Top Van	0.00	0.25	0.25	0.00	0.75	0.25	0.00	4
Passenger Vehicle	0.83	0.83	0.87	0.73	0.77	0.80	0.80	30
Pickup-Utility-Service	0.81	0.71	0.76	0.73	0.79	0.79	0.79	94
Pickup-Utility-Service wTrailer	0.79	0.67	0.88	0.75	0.71	0.88	0.83	24
Platform wTrailer	0.80	0.77	0.80	0.80	0.83	0.83	0.83	30
Platform (SU)	0.87	0.86	0.90	0.86	0.84	0.90	0.89	135
Platform (Semi)	0.91	0.93	0.93	0.91	0.91	0.94	0.94	160
Stake Body (SU)	0.81	0.88	0.87	0.90	0.83	0.89	0.89	114
Tank (Multi)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	5
Tank (SU)	0.64	0.55	0.73	0.55	0.64	0.73	0.64	11
Tank (Semi)	0.94	1.00	0.99	0.96	0.99	0.99	0.99	83
Tank Tank	1.00	0.96	1.00	1.00	1.00	1.00	1.00	27
Others	0.29	0.37	0.45	0.47	0.22	0.37	0.18	49
Accuracy	0.92	0.93	0.94	0.93	0.93	0.94	0.94	3,084
Avg CCR	0.84	0.86	0.88	0.85	0.87	0.88	0.86	3,084

Note: SU: Single-Unit Truck; Semi: Tractors pulling Semi-Trailer; Multi: Tractors pulling multiple trailers. Cells labeled with red colors represent CCR lower than 0.80. Green colors highlight the benefits of using the SMA model.

### LiDAR intensity-based Truck Surface Characterization

Aside from their physical attributes, fleet identification features such as logos found on many trucks can be used to infer their industry affiliation and can serve as another dimension of truck characterization to provide further insights into their activity patterns. Fittingly, along with the depth and geometry information, LiDAR sensors also provide an additional attribute widely named as 'intensity'(I). LiDAR intensity is the measured power (returned) of a reflected laser beam from a target surface. These intensity values are indicator of surface reflectance of the target surface, and are influenced by factors such as range (R), angle of incidence ( $\alpha$ ), target surface roughness, and other instrument related

variables. A significant advantage of LiDAR intensity over conventional images is its independence of ambient lighting, which allows it to be effective in poor lighting conditions, such as twilight and night times. This capability makes LiDAR a candidate technology for capturing truck fleet attributes.

This study focused on estimating the parameters for the LiDAR intensity correction using a data driven technique such as mentioned in (55). As per the proposed framework the corrected intensity values ( $I_c$ ) could be written as a function of observed intensity ( $I_o$ ), range ( $R$ ), and cosine of angle of incidence ( $\alpha$ ) as shown below.

$$I_c = I_o R^a \cos^b(\alpha)$$

Each LiDAR scan of truck is a pointcloud object depicting the cartesian coordinates of truck geometry in 3D. Range and angle of incidence of each of those points needs to be calculated for estimating the Intensity correction parameters. The range of each of the points can be estimated directly by estimating the length of their position vector from the LiDAR scanner itself.

Preliminary results presented in Figure 4 show an improved distinction of fleet identification features. The corrected intensity values show a reduction in the variance of intensity values quite well and provides a distinct contrast of fleet features against the background.

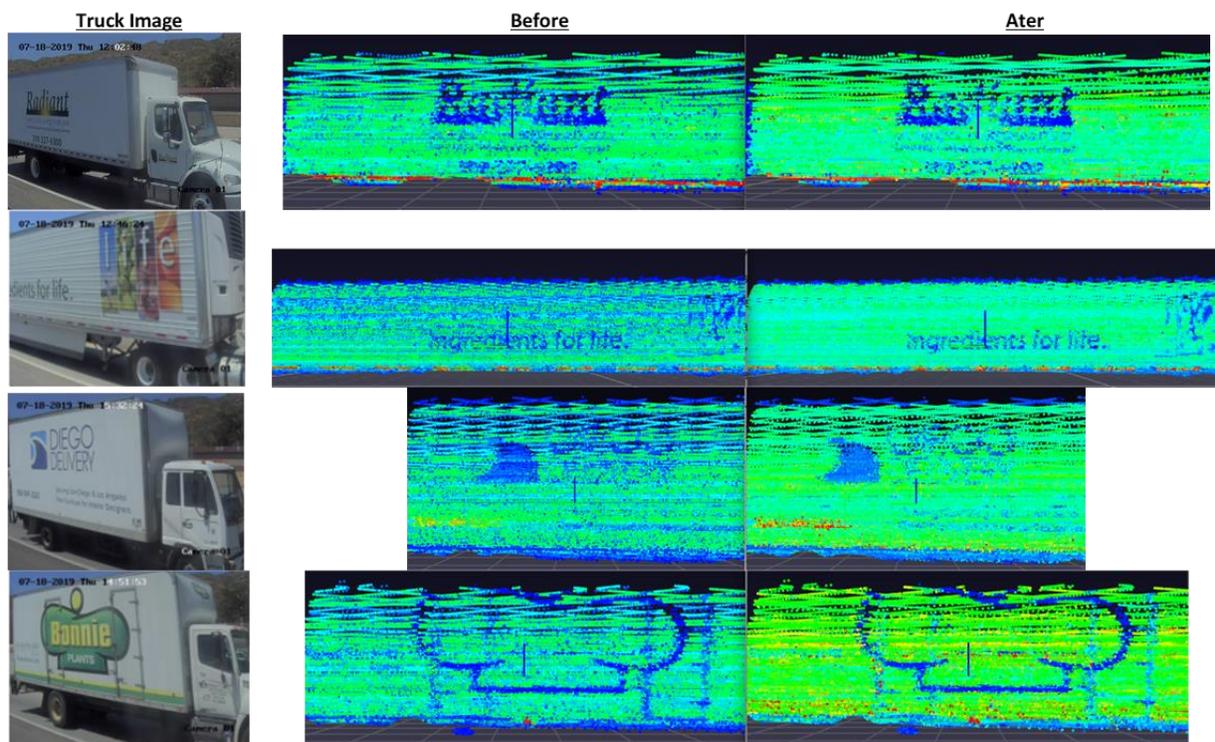


Figure 4. Preliminary results of LiDAR intensity correction

## Conclusion

To fill the truck monitoring gaps on rural highway corridors, this study developed two novel LiDAR-based truck classification methods through the development of a new truck point cloud reconstruction framework that was able to retain a wide LDZ and accurately classify trucks based on the FHWA-CA scheme and detailed truck body configurations. The data used for modeling was collected from a horizontally oriented multi-array 3D LiDAR sensor, which has the ability to capture a wide field of view of the roadway. The horizontal implementation is particularly useful in rural two-lane highways, where the wide field of view can overcome partial occlusion events when a vehicle in the near lane momentarily occludes another vehicle traveling in the opposite direction on the far lane.

The sparse point clouds from individual frames resulting from a low vertical resolution were enriched by aggregating multiple frames associated with the same truck.

The axle-based classification model with the reconstruction framework outperformed the state-of-the-art axle-based classification model using LiDAR sensors both in terms of accuracy and robustness. This LiDAR-based FHWA model achieved a 79 percent average CCR. Classes 8 and 9 were classified correctly with 84 percent and 99 percent CCR even though they share very similar body configurations.

This study investigated the PointNet deep representation learning algorithm to further classify trucks in their detailed body configurations. The LiDAR-based truck body type classification model was able to classify heavy-duty trucks in much more detail, with a close relationship to their industry affiliations. This model was able to classify 31 different vehicle types (advantageously mainly trucks) and achieve an average class CCR of 90 percent for both a truck with trailer (s) and single-unit vehicles. Remarkably, the proposed method was able to distinguish 53ft containers and semi-trailer enclosed vans with over 95 percent CCR even though they share very similar physical characteristics, which is a significant improvement over previous models using the integration of WIM and inductive signature data (38), as well as LiDAR (8).

# 1. Introduction

Heavy trucks comprise a much larger proportion of overall traffic in rural highways compared with their urban counterparts, hence detailed classification counts are needed to adequately assess the impacts of truck activity in these regions. The traffic monitoring guide recommends that at least 30 percent of traffic data collection sites provide classification data. However, piezo-based sensors, which are the predominant technology for collecting truck classification data are costly to maintain. Furthermore, while significant permanent traffic detector infrastructure such as inductive loop, piezo-based axle classifiers, and weigh-in-motion (WIM) systems have been invested in urban regions and major interstate corridors, traffic data along many rural highway corridors are still primarily obtained via temporary pneumatic hose sensors.

Under current practice, truck count data obtained along non-detectorized rural highway corridors are either estimated using unreliable growth factors applied on decades-old observed data or collected via pneumatic tubes which need to be laid across highways to collect traffic data. The former approach is inaccurate and tends to underestimate the actual traffic growth in California. The latter approach exposes field personnel to significant safety hazards and is strongly discouraged unless lane closures are performed. However, lane closures are costly and labor-intensive, which significantly affect the efficiency of data collection efforts. These further limits the number of sites where actual data can be collected in place of estimates due to labor and time constraints. Notwithstanding, reliable rural highway data is still required to obtain necessary federal funding allocation through the Highway Performance Measurement System (HPMS) to ensure adequate maintenance of the rural highway network and monitor the performance of freight activity in these regions.

The rapid advancement of Light Detection and Ranging (LiDAR) technology in recent years provides further opportunities for non-pavement intrusive alternatives to collect detailed vehicle classification data. In this study, a new truck classification method is developed using a LiDAR sensor array in a horizontal orientation, utilizing a reconstruction procedure that combines frames of sparse point clouds to generate a dense point cloud representation of vehicle objects to facilitate accurate truck classification, while preserving the panoramic LiDAR Detection Zone (LDZ). First, vehicle point clouds were extracted by removing the background and clustering the residual points into objects. Then, a new vehicle reconstruction framework was built to enrich the sparse point cloud obtained from the horizontally oriented sensor. Objects associated with the same vehicle from consecutive frames were grouped and combined to generate a dense 3D point cloud representation of each vehicle. Subsequently, the lower profile of the reconstructed vehicle point clouds was extracted and used to classify vehicles based on the FHWA-CA scheme. In contrast to previous studies, which used a classic machine learning framework, this study adopted the PointNet deep representation learning algorithm to train the classification model from the preprocessed point cloud data to classify trucks according to their detailed body configurations. Both classification models were found to produce promising prediction results.

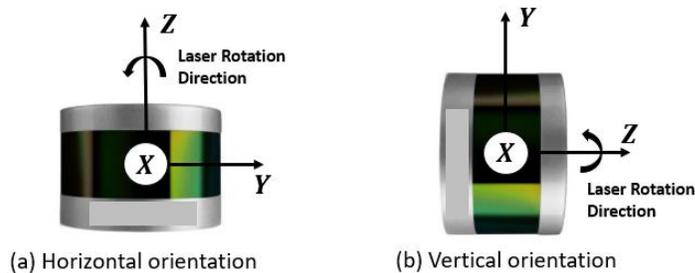
## 2. Literature Review

LiDAR technology was initially investigated for vehicle classification applications in the early 2000s when scanning laser sensors were available for the domain of traffic surveillance. Such sensors scan the cross-section of the roadway by taking several range measurements and generate gray-level intensity images for each vehicle passing through the scanning area. Abdelbaki et al used two laser scanners with a 10-degree separation to classify vehicles based on their aggregated bodies (1). In their study, high-level features such as vehicle length, vehicle width, vehicle height, and speed features were extracted from the intensity images and a rule-based classification lookup table was created to assign vehicles into their corresponding classes based on extracted features. Later, Hussain et al. adopted the same data collection setup with an additional feature, the average of the percent of edge points between two consecutive images, to further improve the classification accuracy (2). Instead of using a rule-based classification method, they constructed a random neural network model for the classification purpose. The prediction error was reduced with their improved classification method (2). Similarly, Sandhawalia et al. interpreted the 3D measurements acquired from a SICK laser scanner as a 2D image, where the pixel intensities were used as the depth values. The vehicle classification problem was posed as an image categorization problem (3). Instead of directly using geometrical vehicle attributes from the profile image, Sandhawalia et al utilized the Fisher vector representation of the profile image, where a set of low-level local features obtained from the profile image was transformed into a high-level image representation (3). Subsequently, high-level feature vectors were extracted from the fisher image signatures and were classified using the one-versus-all linear classifier. This model was able to classify vehicles into 6 categories with an average accuracy of 82.5%. The three aforementioned studies involved overhead sensor mountings which offer the capability to capture detailed depth measurements of each passing vehicle. However, overhead installations are subject to availability of suitable infrastructure. Hence, alternative orientations such as roadside LiDAR setups were explored in subsequent vehicle classification studies.

Lee and Coifman adopted side-fire LiDAR for vehicle classification (4) where they designed a prototype data collection system consisting of a roadside probe vehicle equipped with two vertically oriented laser scanners. Both sensors scan the vertical planar of the road section simultaneously to construct a 3D LiDAR image by merging successive 2D frames (5). After obtaining the 3D point cloud for the surveillance area, the vehicle objects were extracted from the background using a well-established background subtraction algorithm derived from the domain of image processing. Then, six high-level features describing the physical characteristic of each vehicle were extracted from the vehicle cluster as inputs to the classification model. Finally, the vehicle clusters were classified using a decision tree classifier into six distinct classes refined from the length-based classification scheme. Researchers also explored using low-cost single-beam side-fire LiDAR to obtain truck body information. Asborn et al. grouped the raw distance measurements from the LiDAR sensor over time to build vehicle signatures and adopted a Bayesian combined predictor to classify trucks based on their aggregated body type classes (6). They demonstrated cost-effective roadside setup of the LiDAR sensors for collecting classification data. However, neither the laser scanner nor the single-beam LiDAR sensor were designed to provide a detailed vehicle profile, which limited the classification accuracy as well as the diversity of vehicle configurations that could be classified.

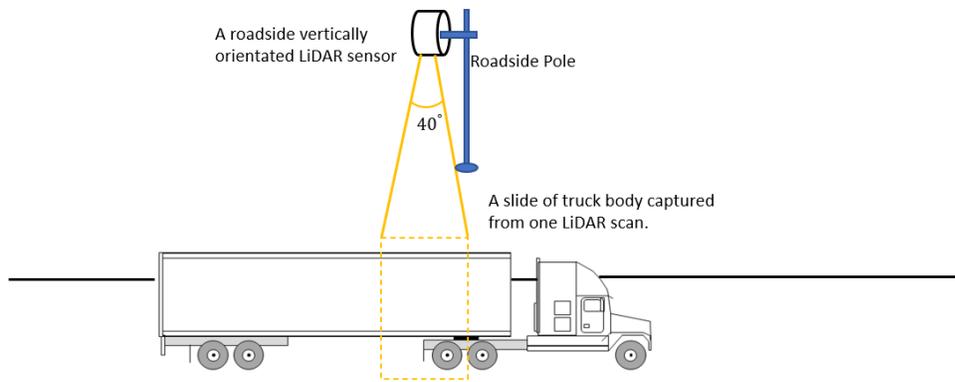
In recent years, multi-array rotating 3D LiDAR sensors have grown in popularity due to the sensing needs of autonomous vehicles. Nezafat et al mounted such a sensor on a roadside pole with a vertical orientation (7), as illustrated in Figure 2.1b.

**Figure 2.1 Illustration of LiDAR Orientation**



When a truck enters the LiDAR detection zone, each scan of the sensor can capture a 3D profile of one side of the truck body within the scanning area. However, the vertical orientation of the sensor constrained the detection zone to a 40-degree horizontal of view (Figure 2.2).

**Figure 2.2 Truck Point Cloud Collection from Vertically Oriented LiDAR**



Therefore, they had to merge all the frames associated with the same vehicle to generate the full profile of a truck. Then, they projected the 3D profiles of trucks to 2D images and adopted a pre-trained convolution neural network model to extract low-level features from the images. Such an image-based method was able to distinguish only 4 types of trucks with a similar configuration, yielding over 95% accuracy. However, the 3D information from the point cloud was not well-utilized in their research, which limited the total number of truck types that could be classified. Adopting the same data processing strategy with the vertical orientation of the sensor, Sahin et al. utilized the 3D profile of the truck obtained from the merged frames to classify truck trailers in detail. Sahin et al. divided the 3D truck profile into six equal-sized rectangular voxels and extracted high-level features that represented the physical characteristics of different trailer types (8). Finally, five different classic matching learning algorithms were explored to classify 9 different trailers with the highest median accuracy of 94.2%. The vertical orientation of the multi-array rotating 3D LiDAR was able to capture a dense representation of each vehicle. However, the narrow detection zone required an assumption of constant vehicle speed to accurately reconstruct the individual frames, which restricts its feasibility under congested traffic conditions where significant acceleration and deceleration may be observed. Wu et al. utilized a

horizontally oriented LiDAR sensor, which provided a 360-degree view of the ambient environment, for vehicle classification (9). Unfortunately, the sparse point cloud representation retrieved from the horizontally oriented LiDAR gave insufficient information for detailed truck classification. Therefore, they were only able to distinguish three different types of trucks (Table 2.1). The evolution of LiDAR-based vehicle classification is listed in Table 2.1.

**Table 2.1 Summary of LiDAR-based Vehicle Classification**

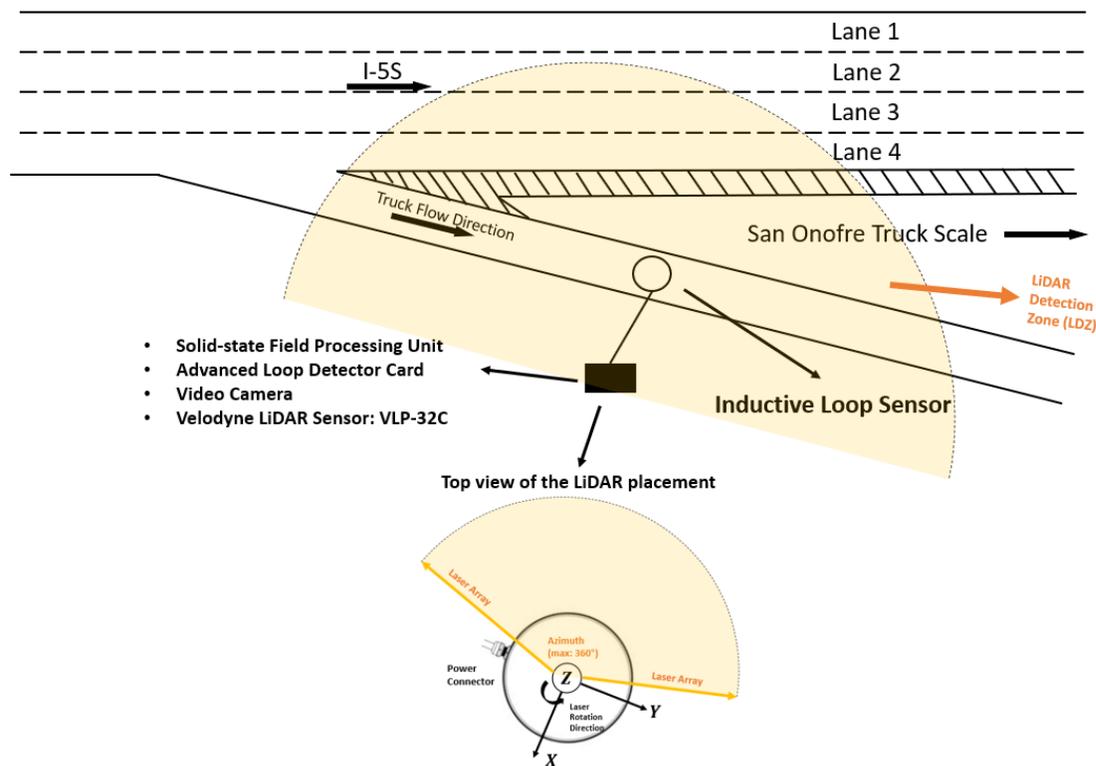
Year	Literature	LiDAR Type	Setup	Methods Type	Classification Method	Correct Classification Rate
2000	Abdelbaki et al., (2001)	Laser scanner	Overhead mounted, two lasers	High-level Hand-designed features from Intensity Image	Rule-based Lookup table	Motorcycle: 66.6%; Passenger vehicle: 87.2%; Pickup/Van/Sport Utility: 90.3%; Misc. Truck/Bus/RV: 84.7%; Tractor Trailer: 100.0%
2005	Hussain & Moussa, (2005)	Laser scanner	Overhead mounted, two lasers	High-level Hand-designed features from Intensity Image	Classic machine learning (random neural network)	Motorcycle: 60.0%; Passenger vehicle: 90.0%; Pickup/van: 94.4%; Single unit truck or bus: 85.0%; Tractor Trailer: 100.0%
2013	Sandhwalia et al., (2013)	Laser scanner	Overhead mounted, each sensor per lane	High-level hand-designed features from raw profile feature, fisher image signatures, side projection profiles	Classic machine learning	Passenger vehicle: 99.8%; Passenger vehicle with one trailer: 89.8%; truck: 81.4%; truck with one trailer: 89.7%; truck with two trailers: 68.8%; motorcycle: 68.7%
2012	Lee & Coifman, (2012)	Laser scanner	Side-fire, vertically orientation, two lidars	High-level hand-designed feature from raw points	Classic machine learning	Motorcycle:91.2%; Passenger vehicle: 99.9; Passenger vehicle pulling trailer: 94.1%; Single unit truck: 94.5%; Single trailer: 68.9%; Multiunit truck: 98.6%
2019	Asborno et al., (2019)	Single beam	Side-fire, horizontal orientation	Combination of High-level hand-designed feature and low-level feature from LiDAR signature	2D LiDAR signature pattern Classic machine learning (Bayesian combined predictor)	Van and container: 94%; Platform type: 63%; Low-profile trailer: 44%; Tank: 33%; Hopper and end dump: 30%
2019	Wu et al., (2019)	Multi-array rotating 3D LiDAR	Side-fire, horizontal orientation	Hand-designed features from raw points	Max height, the nearest distance to lidar, number of points in the frame, the difference between length and height, object profiles Classic machine learning	Bus: 100%; Five-axle, single-trailer truck: 94.1%; Bicycle; motorcycle: 5.9%; Three-axle, single-unit truck: 0%; Passenger car; four-tire, single unit; two-axle, six-tire, single-unit truck: 93.2%; Pedestrians and skateboarder:100%
2019	Vatani Nezat et al., (2019)	Multi-array rotating 3D LiDAR	Side-fire, vertical orientation	Low-level feature from 2D images	Transfer Learning (AlexNet, VggNet and ResNet)	Container: 98.4%; Ref Container: 90.1%; Ref Enclosed Van: 95.7%; Enclosed Van: 97.6%
2020	Sahin et al., (2020)	Multi-array rotating 3D LiDAR	Side-fire, vertical orientation	Hand-designed features from vowelized point cloud	Classic machine learning	20ft Container: 96.3%; 40ft Container: 97.7%; 40ft reefer container:94%; Dry Van: 94.3%; Reefer dry Van: 91.0%; platform: 94.9%; Tank: 97.1%; Auto transport: 91.1; open top and dump: 85.1; other: 62.5

While the side-fire orientation provides advantages in installation setup, it exposes the challenge of capturing detailed dense point clouds. While a vertically oriented LiDAR sensor has the ability to capture dense point cloud over successive frames, its narrow horizontal field of view requires the assumption of constant vehicle speeds. On the other hand, horizontally oriented LiDAR possesses a wider horizontal field of view, but each resulting point cloud is sparse and limits the accuracy of the resulting classification model. Most studies in the literature adopted a similar methodology approach. First, high-level features were selected and derived from either raw or transformed points, such as 2D depth images and voxels. Then, the high-level features were directly used as input for classic machine learning algorithms. The classic machine learning algorithms may work adequately for a classification scheme with a small number of defined features (e.g., the axle-configuration in the FHWA classification scheme). However, for the task of body type classification problem, a combination of a myriad of features potentially can be used to differentiate various truck body types. Therefore, incorporating feature extraction into the optimization process of the classification algorithm would be much more ideal. In summary, it is challenging to collect sufficient information for truck classification while maintaining the view of detection. To address this challenge, this research study investigated a novel truck classification method, which presents a potential for multi-lane truck classification application and is capable of classifying based on both FHWA classification scheme and truck body types in detail with a promising results.

### 3. Data Description and Preprocessing

#### 3.1 Study Site Layout

The data used in this study were collected from the entrance ramp to the San Onofre truck scale from the Southbound I-5 Freeway in Southern California (as shown in Figure 3.1). The LiDAR sensor was placed in horizontal orientation above a traffic cabinet. The z-axis of the LiDAR sensor is aligned perpendicular to the truck flow direction. The yellow sector shown in Figure 3.1 illustrates the approximated LiDAR Detection Zone (LDZ). Data under free flow and congested conditions were observed at the study site and included in the model development.

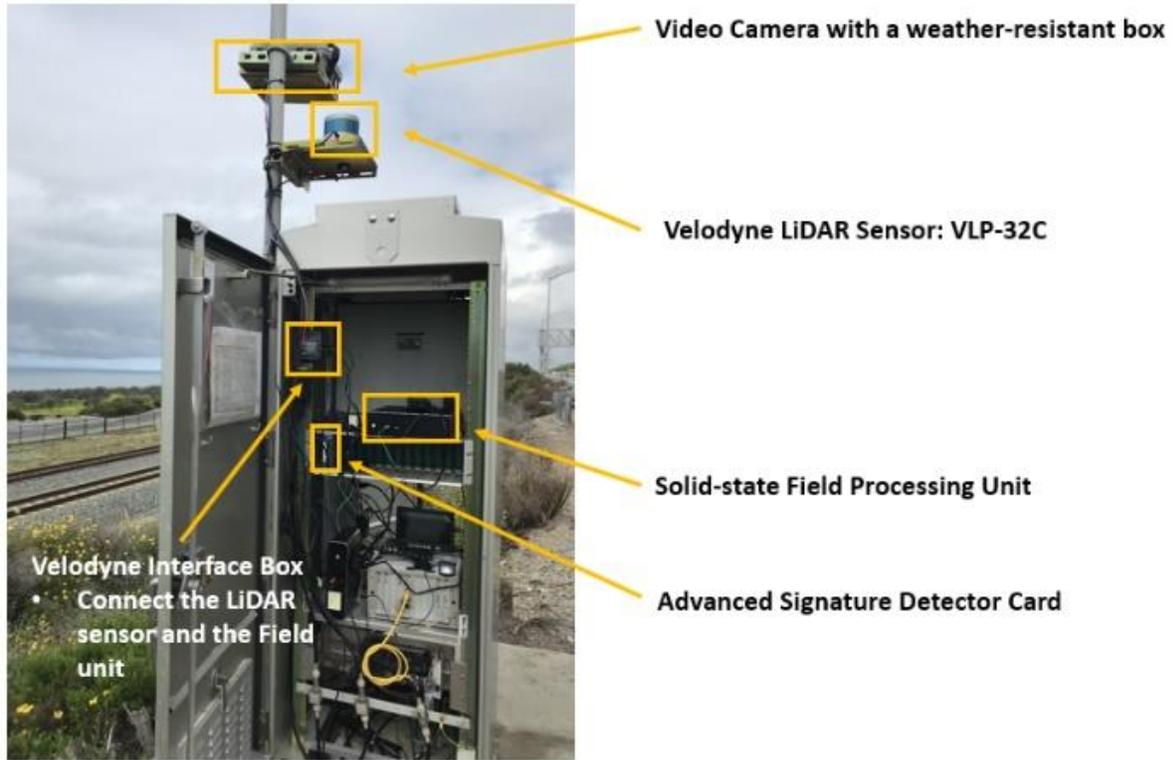


**Figure 3.1** Layout of the Detection Site

#### 3.2 Data Collection Setup

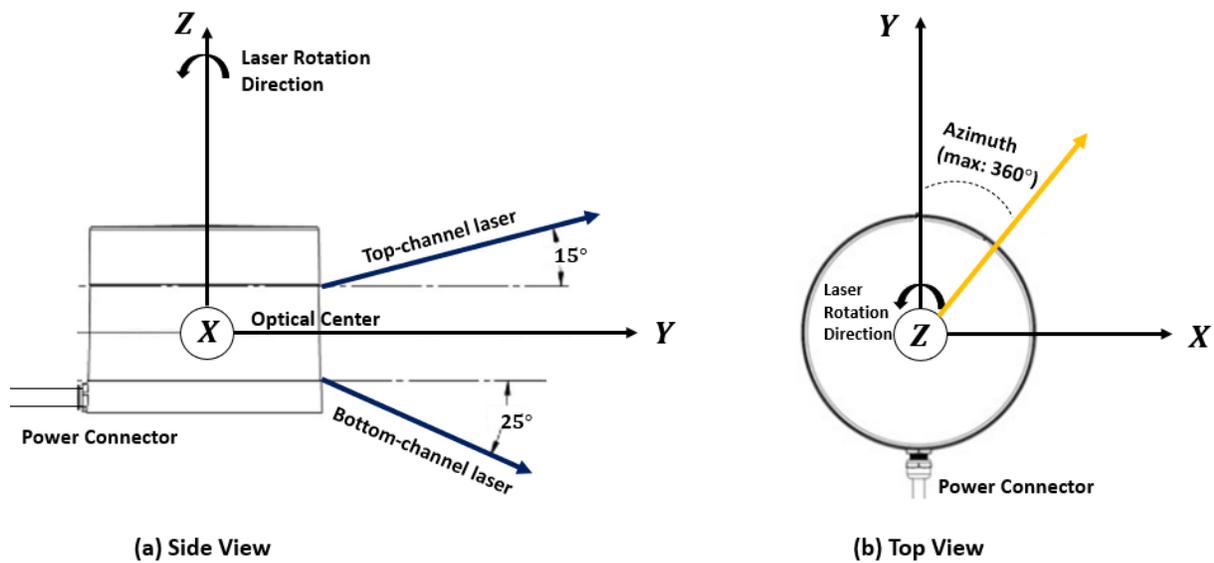
A video camera, an advanced loop detector card and a Velodyne VLP-32c LiDAR unit were installed at the study site as shown in Figure 3.2. The lateral distance between the LiDAR sensor and the centerline of the adjacent traffic lane was approximately 6 meters. All three sensors were connected to a solid-state field processing unit. The video camera and loop detector were used to establish data groundtruth. The Velodyne VLP-32c sensor has 32 infra-red lasers paired with infra-red detectors mounted on a motorized rotating platform to provide distance measurement between the sensor and objects (10).

The LiDAR was configured to scan the surroundings at a frequency of 10 rotations per second with a 180-degree LDZ – each rotation generating a single 3D point cloud frame. The LiDAR sensor was horizontally mounted on a platform attached to the roadside pole of an existing traffic control cabinet.



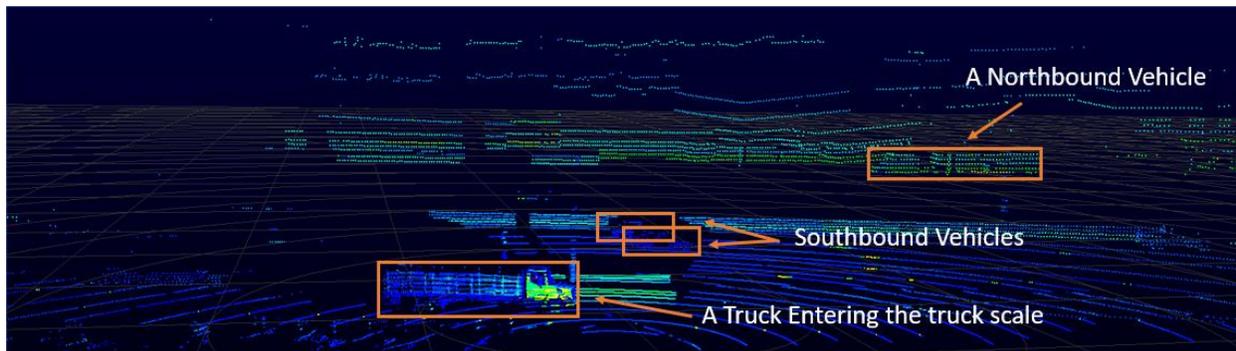
**Figure 3.2 System Setup**

The LiDAR sensor was mounted 2.05 meters above the ground plane and the top laser channel elevation angle was 15 degrees (Figure 3.2), which allowed the sensor to capture both the top and side view of passing vehicles. As shown in Figure 3.2, z-axis is perpendicular to the direction of the traffic and the laser array rotates about the z-axis.



**Figure 3.3 Illustration of the LiDAR Sensor**

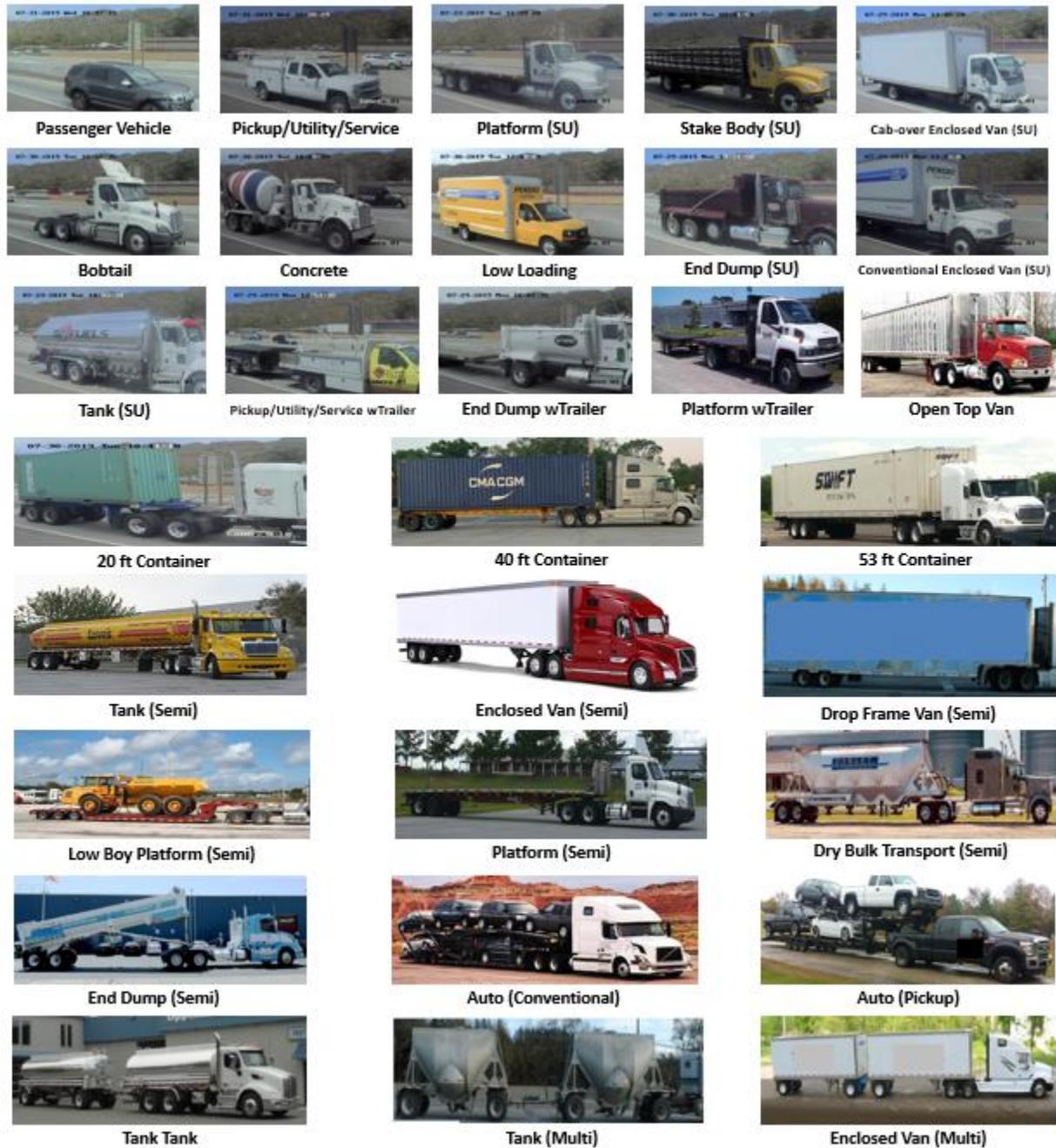
A sample of raw point cloud data frame showing a vehicle entering the truck scale within the LDZ is presented in Figure 3.4.



**Figure 3.4 The Raw Point Cloud of the Detection Region**

### 3.3 Data Description

The video data from the camera, inductive loop signature data from the loop detector, and point cloud data from the LiDAR sensor, were collected simultaneously. Several data collection efforts were made between July 18<sup>th</sup>, 2019 and August 5<sup>th</sup>, 2019, yielding a point cloud dataset comprising 10,024 processed vehicles and representing 30 distinct truck categories defined in this study (including a class labeled as “Other” which represented trucks not belonging to any of the classes shown in Figure 3.5) as well as passenger vehicles (Figure 3.5). 70 percent of the data were used for training and 30 percent were reserved for model testing.



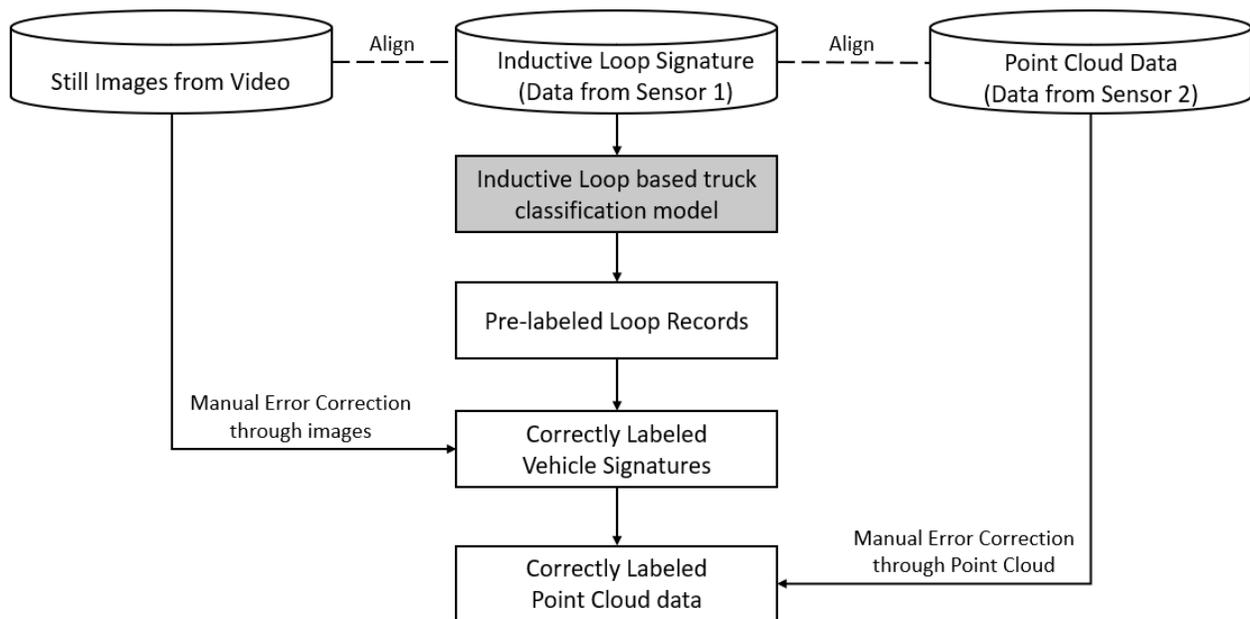
**Figure 3.5 Illustration of Vehicle Body Configuration used in the study (Note: SU: Single Unit Trucks, Semi: Tractors pulling Semi-Trailer, Multi: Tractors pulling Multiple Trailers.)**

### 3.4 Semi-automatic Data Labeling Method

Data labeling is a critical but typically labor-intensive process in vehicle classification modeling research. Conventionally, the data labeling process requires a significant effort to visually determine the class label of each detected vehicle through images from the video camera, and subsequently manually

record the corresponding vehicle characteristics. In this study, a semi-automatic data labeling strategy was developed and implemented to improve the efficiency of the data labeling process and further enrich the training dataset. The semi-automatic data labeling method utilized inductive loop signature classification predictions as a preliminary data labeling mechanism to establish the dataset used to train the LiDAR-based classification model. Note, inductive loop sensors are not required for the LiDAR-based model implementation and calibration.

The overall semi-automatic data labeling process is illustrated in Figure 3.6. First, vehicle records from all three data sources (inductive loop, video and LiDAR) were synchronized and aligned. When a vehicle enters the inductive loop sensor's detection zone, a timestamp is generated with its inductive signature. Next the corresponding images from the video camera are cropped based on the synchronized timestamp. Then, a time window with a range of  $\pm 0.1$  seconds around the time instance that the vehicle hit the loop sensor is created. The vehicle point cloud object which contains frames within the time window are subsequently aligned with the vehicle image. This study adopted a signature-based truck classification model to pre-label vehicle images (11). All vehicle records were still manually verified against their video images. However, processing effort is significantly reduced as only misclassified data required manual label correction. Finally, the labels were applied to corresponding point clouds as the data sources were aligned and integrated. This semi-automatic data labeling strategy significantly accelerated the labeling process for emerging sensors (Sensor 2 in Figure 3.6) with the support of the existing technology (Sensor 1 in Figure 3.6).



**Figure 3.6 Semi-automatic Data Labeling Framework**

### 3.5 Data Preprocessing

#### 3.5.1 Background Subtraction and Object Detection

A large set of points in the raw point cloud was associated with static objects, e.g., ground, buildings. These background points were required to be subtracted from the raw point cloud to reduce the data processing time and improve the accuracy of the proposed classification method.

A point cloud background subtraction method was developed based on spatial occupancy. The algorithm started with aggregating the point cloud data over an initiation period  $t$ . The points originating from each of the 32 laser channels of the LiDAR were distributed on a conical surface, with the LiDAR unit being the apex. The elevation angles of the channels are listed below:

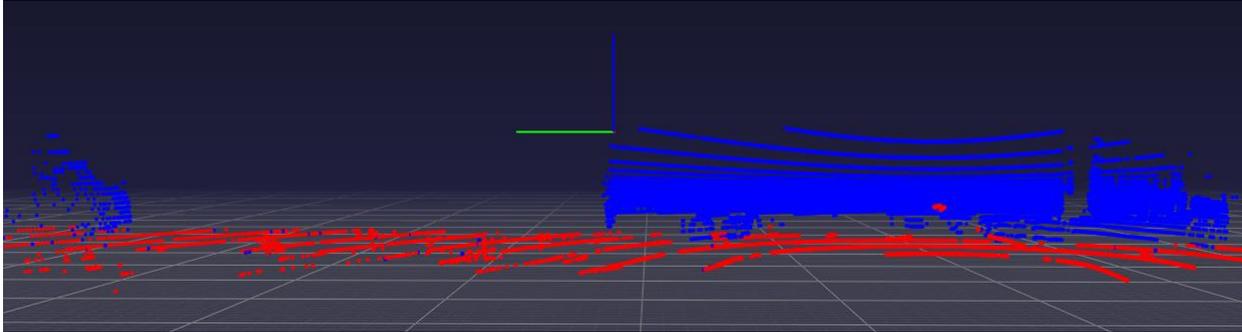
**Table 3.1 Laser Channel Angles**

ID	Elevation Angles (deg)	ID	Elevation Angles (deg)
0	15	16	-1.667
1	10.3333	17	-2
2	7	18	-2.333
3	4.667	19	-2.667
4	3.333	20	-3
5	2.333	21	-3.333
6	1.667	22	-3.667
7	1.333	23	-4
8	1	24	-4.667
9	0.667	25	-5.333
10	0.333	26	-6.148
11	0	27	-7.254
12	-0.333	28	-8.843
13	-0.667	29	-11.31
14	-1	30	-15.639
15	-1.333	31	-25

The cumulative number of points within each cell on the conical surface was subsequently counted, defined by a horizontal angular resolution  $\theta$ , the channel number  $I$ , and the radial distance resolution  $r$ . The cells that were occupied by the background objects were deemed to have more points, which was identified using a threshold  $h$ . These background cells were used as a mask to filter out background points in other frames. The initiation period  $t$  should be long enough to accumulate enough points in the environment, but not too long to waste computational resources. Smaller point/range resolution values led to smaller cells, which depicted the background with better resolution. However, too-small cells led to a much longer initialization time and higher computational complexity. The horizontal angular resolution of the Lidar sensor, which is the angular difference between two adjacent points on the same channel, is also an important factor to consider when choosing  $\theta$ . The Lidar used in this study has an

angular resolution of  $0.2^\circ$ .  $\theta$  should be larger than  $0.2^\circ$  so that at least one point falls into each cell at each frame.

The threshold  $h$  should be tuned based on all other parameters. The following parameters were used in this study:  $t = 60\text{sec}$ ,  $\theta = 1^\circ$ ,  $r = 0.1\text{m}$ ,  $h = 10$ . The background subtraction result of frame 126,010 on July 19, 2019, is illustrated in Figure 3.7, where red represents background and blue represents foreground.



**Figure 3.7 Distinguishing foreground vehicle objects (blue) from the background (blue) to facilitate background subtraction**

To effectively identify individual vehicles, the foreground points were partitioned based on their similarity. Density-based spatial clustering of applications with noise (DBSCAN) is a popular clustering algorithm that generally works well in existing point cloud studies (12). The algorithm groups points based on their proximity. It also marks isolated points as outliers, which makes the algorithm less sensitive to background points that were not removed from the previous step. The algorithm takes two parameters:

- *minPoints*: the minimum number of points to form a cluster. Its value should be close to the point cloud size of a typical vehicle in the dataset.
- *eps*: the searching radius to form a cluster. If *eps* is too small, a large proportion of the points will not be clustered; whereas for a large *eps*, most of the points will be allocated to the same cluster.

Tuning DBSCAN parameters is a sophisticated process. In addition to the heuristic (12) offered by the authors of the DBSCAN, researchers have proposed many different approaches (12, 13) to determine the optimal parameters. To find optimal parameters and to conduct sensitivity analysis for DBSCAN are beyond the scope of our research. Instead, we estimate these parameters following engineering heuristics. We acknowledge that the resulted parameters are sub-optimal but are adequate for the vehicle classification task.

For the test dataset, *eps* should be slightly smaller than jam density spacing such that two adjacent stopped vehicles are not clustered as the same vehicle. However, if *eps* is too large, the truck's cabin could be identified as a separate object from the trailer. We chose *eps* to be 1.5 meters. The physical meaning of *eps* is the maximum distance between two points to be considered as one cluster. It is chosen to be slightly smaller than the jam density such that even in completely stopped traffic, two

adjacent vehicles are not grouped into one cluster. For this reason, it does not need to be tuned under different traffic conditions. The other parameter *minPoints* is estimated based on the size of a typical truck. The value is chosen such that a typical semi-truck can be detected when it is on the off-ramp (measured from the tip of the offramp divider), which is around 60 meters from the sensor. Below is the process to estimate the number of points captured by LiDAR when the distance between the truck's front face and LiDAR is  $d$ . The scenario is illustrated in Figure 3.8 with a side view and top-down view,

where,

$w$ - width of the truck

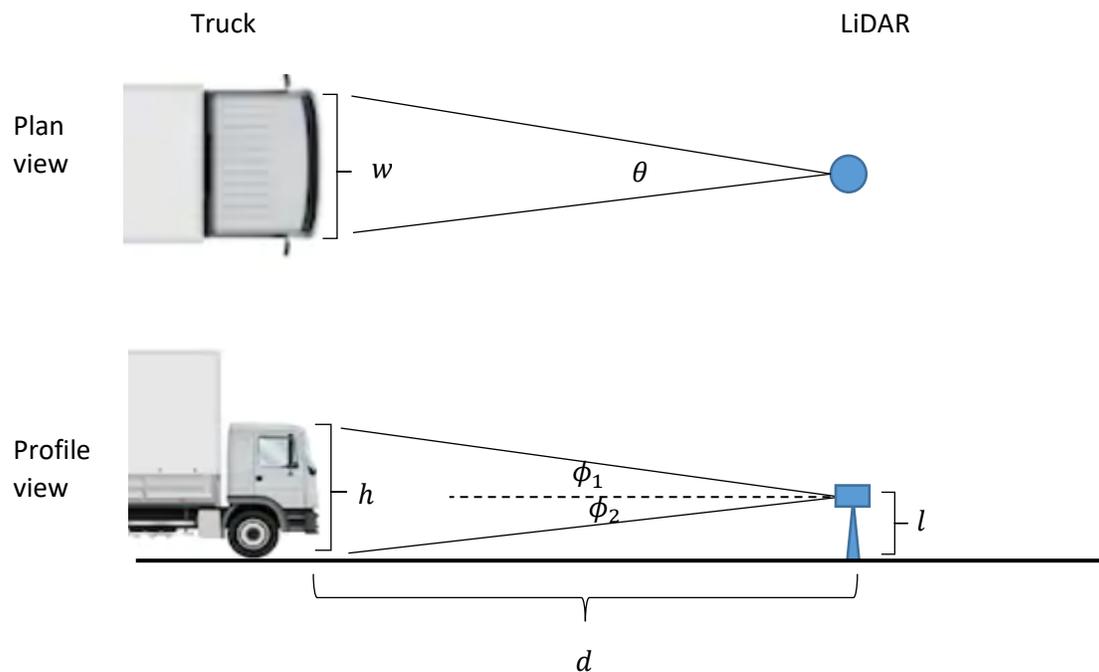
$h$ - height of the truck

$\phi_1$ - angle of elevation from the LiDAR to the truck

$\phi_2$ - angle of depression from the LiDAR to the truck

$\theta$ - beam width of the truck viewed by the LiDAR

$l$ - mounting height of the LiDAR



**Figure 3.8 Detection Illustration**

From the top-down view,

$$a = \rho\theta \approx \frac{\rho w}{d} \quad (1)$$

Where,

$\rho$ - azimuth resolution of the sensor, represented as the number of points within a unit angle.

$\alpha$ - number of points per laser channel.

From the side view,

$$\phi_1 = \arctan\left(\frac{h-l}{d}\right) \quad (2)$$

$$\phi_2 = \arctan\left(\frac{l}{d}\right) \quad (3)$$

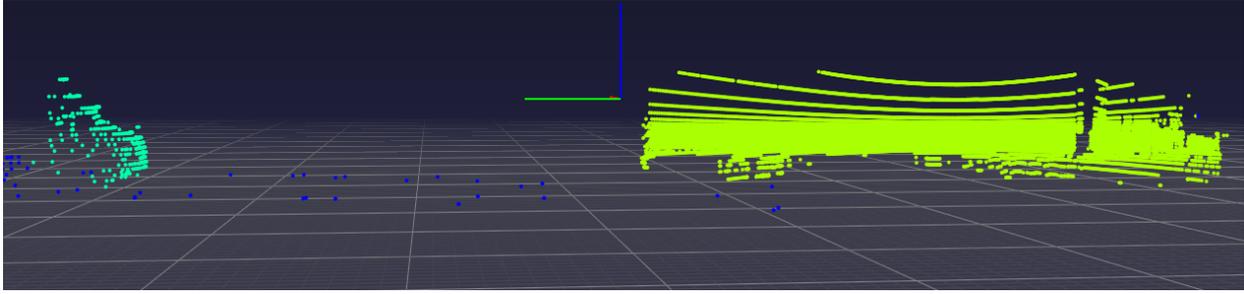
For the VLP-32c LiDAR with a 10Hz sampling rate,  $\rho = 10$  points per degree. The mounting height of the LiDAR  $l = 2.05m$ . The size limit of trucks defined by the California Department of Transportation (Caltrans)<sup>1</sup> is used for the calculation, where  $w = 2.59m$  (8.5 feet),  $h = 4.26m$  (14 feet). Then,  $\phi_1 = 2.11^\circ$ ,  $\phi_2 = 1.96^\circ$ . Based on  $\phi_1, \phi_2$ , and the elevation angles of the laser channels in Table 3.1, the number of laser channels that cover the truck,  $\beta = 10$ . The total number of points captured by the Lidar from the truck's front face at distance  $d = 60m$  is thus,

$$N = \alpha \cdot \beta \approx \frac{\beta \rho w}{d} = 247 \quad (4)$$

To accommodate trucks that are slightly smaller than the typical size. We choose *minPoints* to be 200, which is slightly smaller than  $N$ .

The object detection result of frame 126,010 on July 19, 2019, is illustrated in Figure 3.9. The outliers are represented by blue points, while objects are represented in other distinct colors.

<sup>1</sup> <https://dot.ca.gov/programs/traffic-operations/legal-truck-access/restrict-process>



**Figure 3.9 Object Detection Result (Blue: outliers; Green: vehicle 1; Yellow: vehicle 2)**

### 3.5.2 Data Association

The same vehicle object presents in multiple consecutive LiDAR frames. The vehicle object in each LiDAR frame needs to be labeled with the same vehicle ID. This research utilized the Simple Online and Realtime Tracking (SORT) algorithm to associate the vehicle point cloud from each LiDAR frame to its corresponding vehicle object efficiently (14). First, each vehicle point cloud was represented by the centroid of the minimum oriented 2D bounding box which was obtained from its ground projection. Next, the inter-frame displacements of each vehicle point cloud were estimated using a linear constant velocity model—Kalman Filter (15). Finally, the vehicle point clouds were optimally assigned to their corresponding vehicle object group using the Hungarian algorithm (16). SORT framework was claimed to be capable of handling short-term occlusion caused by passing objects (14).

## 4. Vehicle Point Cloud Registration Framework

### 4.1 Introduction to Point-set Registration

Point-set registration is an essential component that is widely used in the field of robotics and computer vision. It is the process of estimating the spatial transformation (e.g., translation, rotation, and scaling) that aligns two sets of points from the same object with a sensor that captures them from different views. Given two corresponding point sets  $P = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_m\}$  and  $Q = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots, \mathbf{q}_n\}$  in  $\mathbb{R}^d$  ( $d$  represents the dimension of each point. In this study,  $d = 3$ ), the goal of registration is to search for an optimal rigid transformation matrix  $\mathbf{T}_{PQ}$  composed of a rotation matrix  $\mathbf{R}(\theta_x, \theta_y, \theta_z)$  and a translation vector  $\mathbf{t}(t_x, t_y, t_z)$  to match point set  $P$  with point set  $Q$ . The parameters  $\theta_x, \theta_y, \theta_z$  represent the counter-clockwise rotation angle of the point set about the  $x, y, z$  axis, respectively. The values  $t_x, t_y, t_z$  are the translation of the point cloud along the corresponding axis. In a homogeneous coordinate, a transformation matrix  $\mathbf{T}_{PQ}$  that is used to align point set  $P$  and  $Q$  can be expressed as:

$$\mathbf{T}_{PQ} = \mathbf{T}_{PQ}(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (5)$$

The 3D rotation about  $x, y, z$  axis ( $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ ) and translation matrix  $\mathbf{T}$  is shown below:

$$\begin{aligned}
\mathbf{R}_x &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{R}_z &= \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{6}$$

The most classic method used for solving point set registration problems is called the iterative closest point (ICP) algorithm (17). The ICP algorithm starts with the initial transformation matrix  $\mathbf{T}_0 = (\mathbf{R}_0, \mathbf{t}_0)$  and then selects a set of  $k$  corresponding points pairs  $(\mathbf{p}_i, \mathbf{q}_i)$  between point sets  $P$  and  $Q$ . The distance between  $P$  and  $Q$  can be written as:

$$\text{dist}(\mathbf{T}_{PQ}(P), Q) \tag{7}$$

$\mathbf{T}_{PQ}(P)$  represents rotating and translating  $P$  with a transformation matrix  $\mathbf{T}_{PQ}$ .  $\text{dist}()$  denotes the distance between point sets. In the literature, there are two common ways to define the distance between point sets: Point-to-Point (17) and Point-to-Plane distance (18).

### 1. Point-to-Point Distance Evaluation (17)

Assuming there are  $N$  corresponding point pairs  $(\mathbf{p}_i, \mathbf{q}_i)$ ,  $i = 1 \dots N$ , the registration problem using point-to-point distance measurement can be formulated as:

$$\text{argmin}_{\mathbf{T}_{PQ}} \frac{1}{N} \sum_{i=1}^N \|\mathbf{T}_{PQ}\mathbf{p}_i - \mathbf{q}_i\|^2, \quad \text{s.t. } \mathbf{R}^T \mathbf{R} = \mathbf{I} \tag{8}$$

### 2. Point-to-Plane Distance Evaluation (18).

When Point-to-Plane distances are used as the error metric, the objective function can be formulated as the sum of the square error between  $\mathbf{p}_i$  and the tangent plane at  $\mathbf{q}_i$ . The norm of the tangent plane at  $\mathbf{q}_i$  is denoted as  $\mathbf{nor}_i$ . The objective function is shown below:

$$\text{argmin}_{\mathbf{T}_{PQ}} \frac{1}{N} \sum_{i=1}^N \|(\mathbf{T}_{PQ}\mathbf{p}_i - \mathbf{q}_i) \cdot \mathbf{nor}_i\|^2, \quad \text{s.t. } \mathbf{R}^T \mathbf{R} = \mathbf{I} \tag{9}$$

The next step of the ICP algorithm is to iteratively find the optimal  $\mathbf{T}_{PQ}$  which minimizes the distance between  $P$  and  $Q$ . Due to the simplicity of the original algorithm, hundreds of ICP-based variants have been proposed over the past two decades; a comprehensive review of ICP-based methods has been documented in (19).

## 4.2 Probabilistic Point-set Registration

However, the performance of ICP-based approaches suffers from the noisiness, outliers, and occlusions of point clouds that commonly occur in a real-world dataset, especially for data collected in outdoor

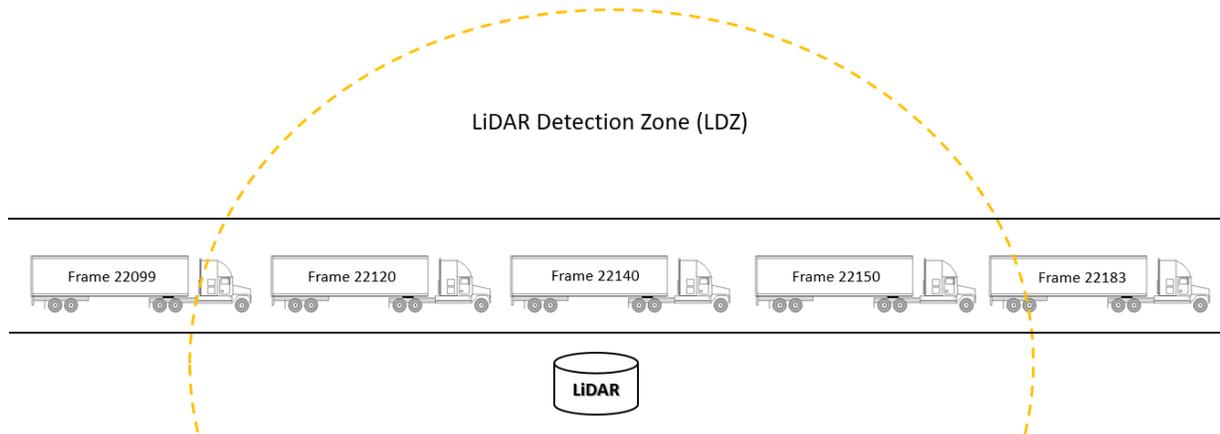
environments (20)(21). Many researchers have investigated probabilistic approaches to improve the robustness of point-set registration. The most popular probabilistic-based registration algorithm is called Coherent Point Drift (CPD) proposed by Myronenko and Song, which treats registration as a probability density estimation problem (21). Instead of using the closest distance to define the corresponding point pairs, CPD assigned a probability value to the correspondence according to the proximity between points from two pointsets. Several studies followed the path of investigating new probabilistic approaches to further enhance the robustness of the registration algorithms (22–24). Unfortunately, such approaches gain robustness while compromising the computation efficiency, limiting their application to large datasets. Gao and Tedrake developed a computationally efficient probabilistic-based registration model—FilterReg—which adopted Gaussian filtering methods to enhance the model efficiency as well as to preserve the robustness and accuracy of the registration process (25). FilterReg has been demonstrated to be computationally faster than the modern ICP variants (25). Therefore, this research adopted the FilterReg algorithm to estimate transformation matrices between consecutive frames.

### 4.3 Vehicle Point Cloud Registration Framework

Most of the previous research on point-set registration targeted aligning point sets obtained from mobile sensors, where the LiDAR unit is mounted on the top of a moving robot (19) which allows the sensor to actively capture the object point clouds. As a consequence, every point cloud density associated with the same object is relatively uniform across LiDAR frames. However, for traffic surveillance applications, LiDAR sensors are generally mounted on a static pole standing by the roadside to passively monitor roadway traffic. As a vehicle traverses the LDZ, the density of its point cloud will gradually increase and then decrease due to its proximity to the sensor. Therefore, this study modified the existing point-set registration framework to better adapt to the data characteristics of vehicle point clouds collected from roadside LiDAR sensors, and then to provide promising vehicle point cloud registration results to support the needs of FHWA axle-based vehicle classification.

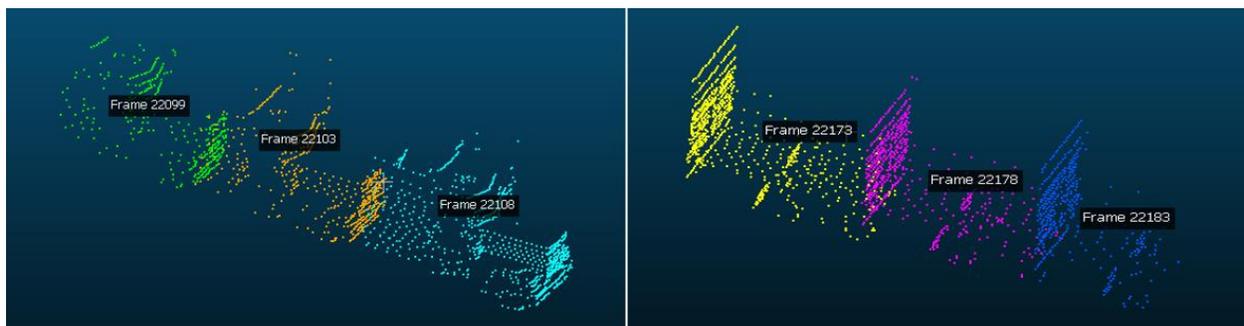
#### 4.3.1 Eliminate Redundant Frames

When a truck is entering or leaving the LDZ (Figure 4.1), the position of the truck is far from the LiDAR sensing unit which results in a sparse truck point cloud (Figure 4.2).



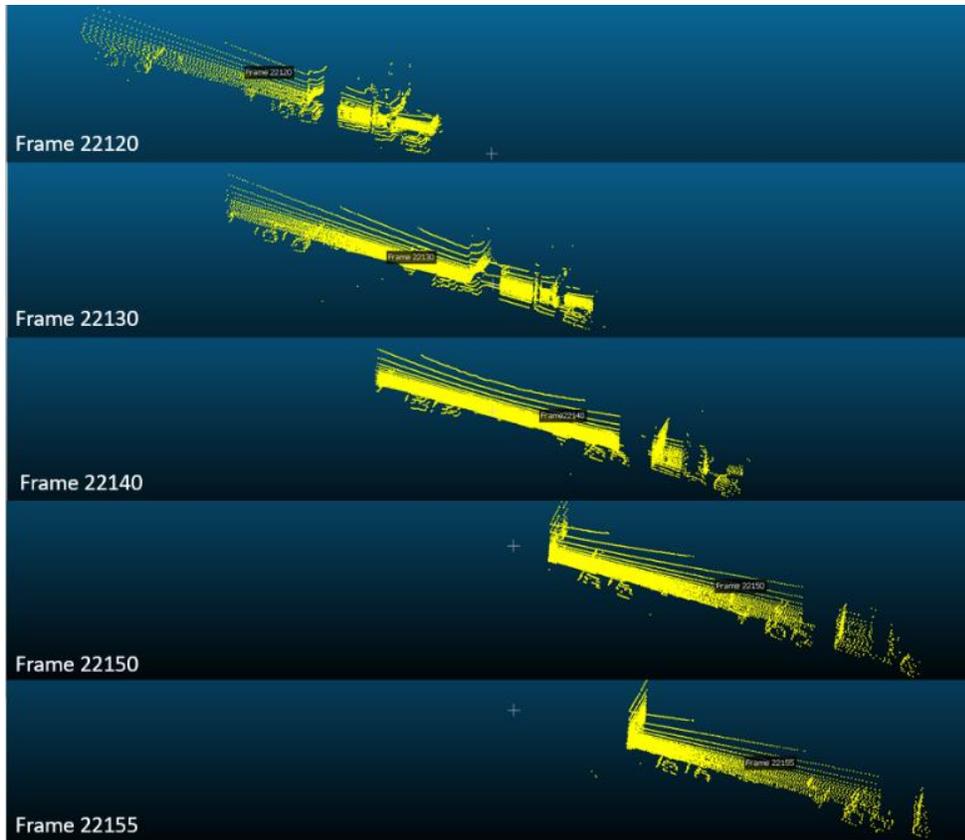
**Figure 4.1 Samples of Truck Frames**

Those frames generally describe the driving unit and the rear edge of the truck as shown in Figure 4.2.



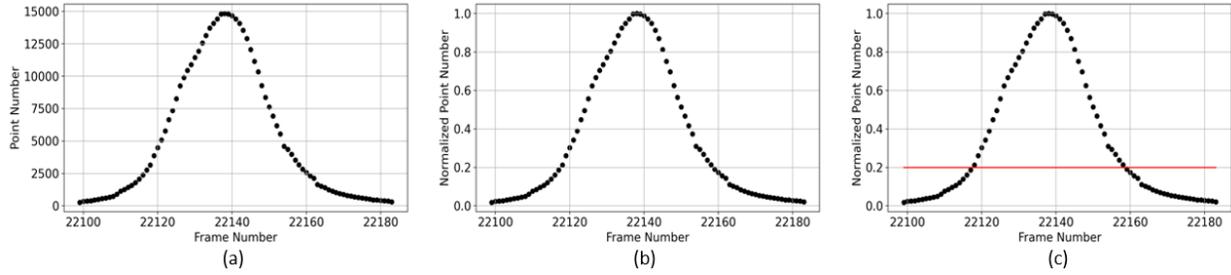
**Figure 4.2 Samples of Redundant Frames**

Such information is captured in the frames associated with the truck when its position is closer to the sensor (Figure 4.2). Thus, the sparse point clouds which are captured far from the sensor and have limited contribution to the registration process are eliminated to save the computation time.



**Figure 4.3 Samples of Frames used for Registration**

Figure 4.4a presents the point counts profile while the truck is traversing the LDZ. Each point in the profile records the total number of points that the truck contained in its corresponding frame. Frame 22,138 and Frame 22,139 contain the highest number of points across all frames during its travel in the LDZ where the highest point count is denoted as  $pn_{max}$ . The point count profile is subsequently normalized based on  $pn_{max}$  (Figure 4.4b). Finally, the truck point cloud which contains point counts less than 20 percent of  $pn_{max}$  were treated as redundant frames and eliminated.



**Figure 4.4 Elimination of Redundant Frames**

### 4.3.2 Statistical Outlier Removal and Voxel Down Sampling

After the background subtraction step is performed, there would still be some noise and outlier data points which are statistically detectable. Therefore, an outlier removal process is needed prior to vehicle point cloud registration. Two procedures are included in this step: statistical outlier removal and voxel downsampling. These are suggested by a popular 3D data processing library—Open3D (26). The statistical outlier removal method takes the 50 nearest neighbors of a given point in the point cloud and considers the points which are 2 standard deviations away from the given point as statistical outliers. Next, in order to increase the computational efficiency as well as preserve the structure of point clouds, point clouds are further uniformly downsampled using a voxel downsampling approach, where points are bucketed into voxel with a size of  $vs\_pre = 0.01$  meter and represented by a single point calculated through averaging all points within the voxel.

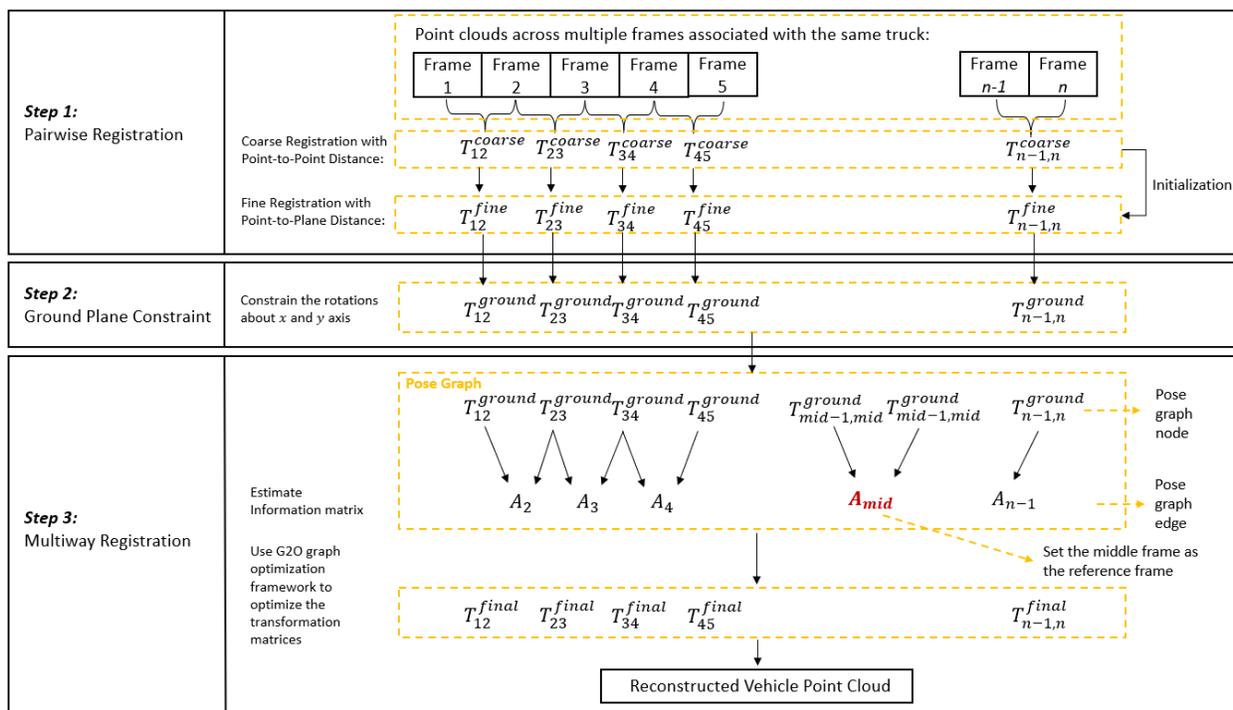
### 4.3.3 Vehicle Point Cloud Registration

After the redundant frames and statistical outliers were removed, a pairwise registration with a coarse-to-fine strategy was applied on each pair of adjacent frames. The pairwise alignment was accomplished through the use of the FilterReg method (25). First, a coarse registration was initially conducted, where all point clouds were coarsely downsampled with relatively larger voxel size  $vs\_coarse = 1.5$  meters and then each pair of point clouds was aligned based on the point-to-point distances metric. Transformation matrices were saved and denoted as  $T_{j-1,j}^{coarse} = [T_{12}^{coarse}, T_{23}^{coarse}, T_{34}^{coarse}, \dots, T_{n-1,n}^{coarse}]$ , where  $j$  is the frame index. Second, the  $T_{j-1,j}^{coarse}$  was fine-tuned using point-to-plane distances with voxel size  $vs\_fine = 0.015$  meter. The transformation matrices obtained from fine registration was written as  $T_{j-1,j}^{fine} = [T_{12}^{fine}, T_{23}^{fine}, T_{34}^{fine}, \dots, T_{n-1,n}^{fine}]$ . Since the basic assumption about vehicle point clouds is that all the point clouds associated with the same vehicle should land on the same plane, the vehicles will not rotate along the  $x$  and  $y$ -axis. Hence, the transformation matrices were constrained on  $x$ - and  $y$ -axis rotation, where the corresponding elements in the matrices were set to zero, as shown in Equation 10:

$$T_{j-1,j}^{ground} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & t_x \\ \sin\theta_z & \cos\theta_z & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

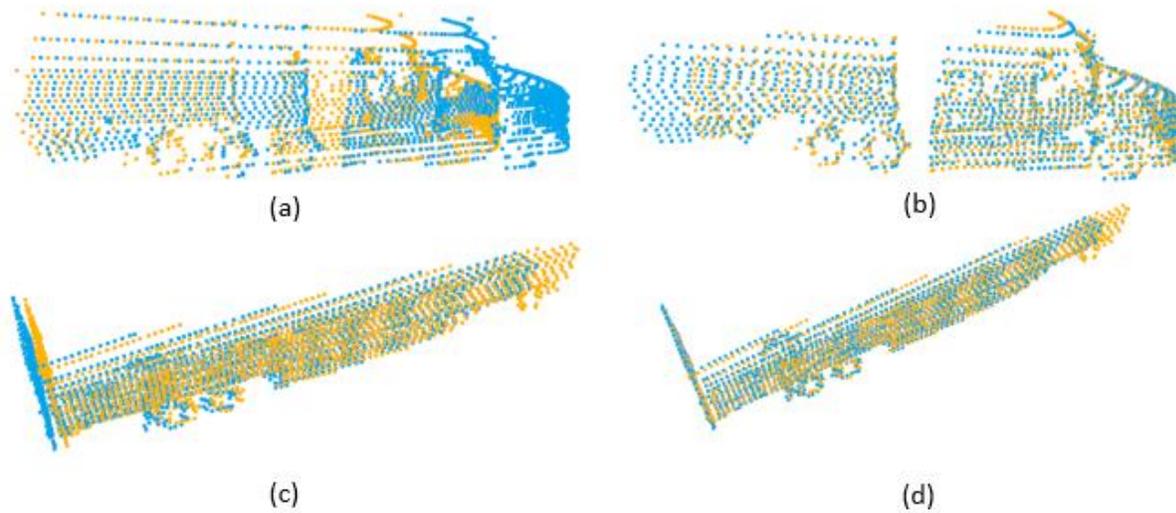
Third, in order to reduce the cumulative errors which could be potentially caused by the sequential pairwise registration, the transformation matrices were further optimized using the multiway registration which describes the process of merging multiple frames of an object in a global space. In this

study, multiway registration was implemented through the use of a pose graph optimization technique proposed in (27). The multiway registration process is illustrated as follows. First, the information matrices which represent the inverse correlation matrix between two consecutive transformation matrices were estimated. Second, a pose graph is defined with the transformation matrices ( $T_{j-1,j}^{ground}$ ) as the node and information matrices ( $A_{n-1}$ ) as the edges in the graph, where each edge of the pose graph connects two nodes. The middle frame of the vehicle object was set to be the reference frame with index  $j = mid = \text{ceil}(\frac{n}{2}, 0.5)$ . All frames were aligned to the reference frame during the optimization process. The pose graph is optimized using the G2O graph optimization framework (27). The final transformation matrices that were used to reconstruct the vehicle point cloud were  $T_{j-1,j}^{final} = [T_{12}^{final}, T_{23}^{final}, T_{34}^{final}, \dots, T_{n-1,n}^{final}]$ . The overall vehicle point cloud registration framework is shown in Figure 4.5.



**Figure 4.5 Vehicle Point Cloud Registration Framework**

When the vehicle is approaching the LiDAR sensor, most of the information is captured from the tractor unit. The distinctive details as well as the level of the sparseness of the point cloud on the truck tractor make the process of finding corresponding points between two point clouds easier. Hence, minimizing the point-to-point distance is capable of aligning the source (Yellow in Figure 4.6) to the target point cloud (Blue in Figure 4.6) firmly. Figure 4.6a presents the point clouds from two consecutive frames. Figure 4.6b is the result of the coarse registration with point-to-point distance.



**Figure 4.6 Examples of Pairwise Registration (Blue: target point cloud, Yellow: Source point cloud)**

On the contrary, while the truck is just passed the LiDAR sensor, points are densely distributed on the side view of the truck. Such point clouds contained a limited number of prominent features to align them by just minimizing the point-to-point distance. Figure 4.6c shows the failure case after coarse registration using the point-to-point distance. However, the dense point distribution on the truck sides creates well-defined planes which allow the fine registration with a point-to-plane strategy to successfully further tighten two point clouds (Figure 4.6d).

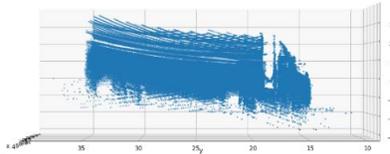
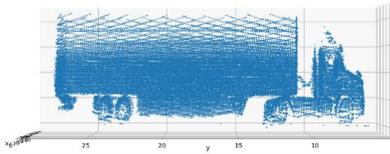
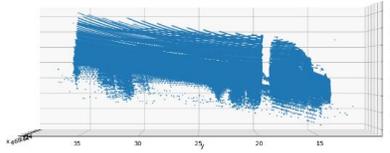
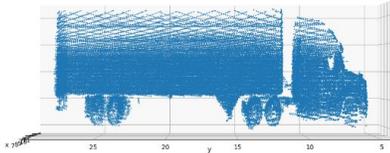
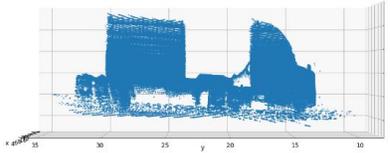
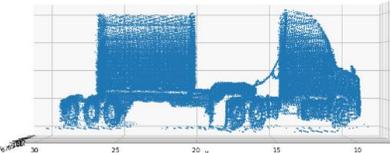
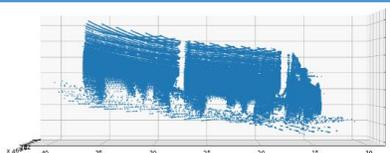
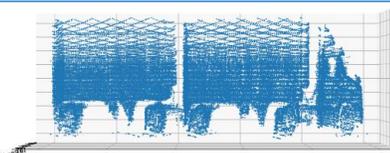
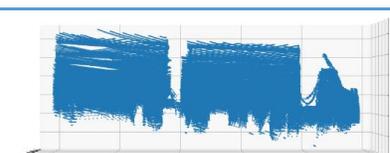
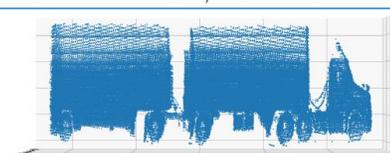
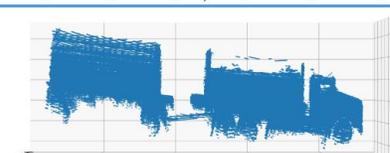
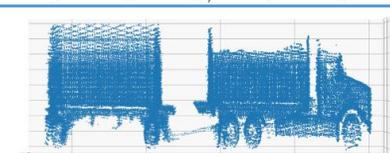
#### 4.3.4 Registration Framework Comparison

The main purpose of the vehicle registration process is to enrich the information of the vehicle point clouds through merging multiple frames and to precisely portray the vehicle characteristics such that vehicles can be classified in detail. Therefore, if the reconstructed vehicle contains essential features which can be used to visually identify its vehicle class without any significant misalignment, it will be considered as a well-registered vehicle point cloud. Otherwise, it will be considered as a poor-registered vehicle point cloud. In this study, the vehicle point clouds registration precision (VPCRP) is defined as:

$$VPCRP = \frac{N_{wr}}{N_{wr} + N_{pr}} \quad (11)$$

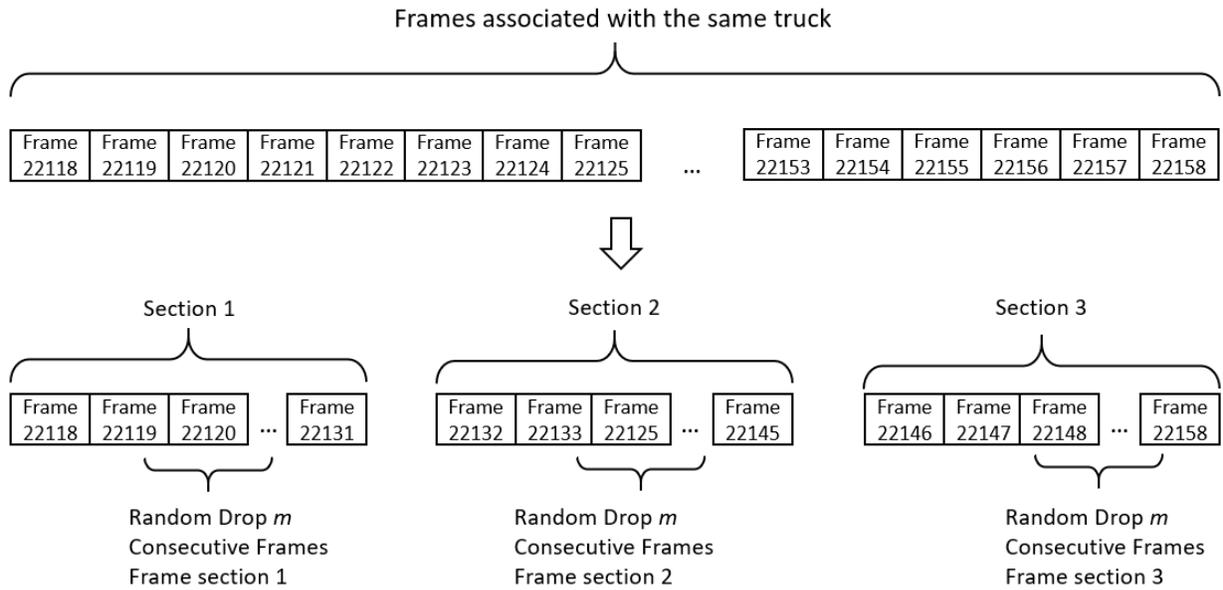
Where  $N_{wr}$  represents the number of well-registered vehicle point clouds and  $N_{pr}$  is the number of the poor-registered point clouds. The new framework has been compared with the previous registration framework through VPCRP value. The previous registration framework (28) presents a VPCRP value of 0.168, where the new registration framework has a VPCRP value of 0.024. The registration performance has been significantly improved by adopting the new framework proposed in this study (Table 4.1).

**Table 4.1 Registration Results Comparison**

FHWA-CA Classes	Previous Registration Framework (28)	The New Registration Framework
Class 8		
Class 9		
Class 10		
Class 11		
Class 12		
Class 14		

#### 4.3.5 Registration Performance with Missing Frames

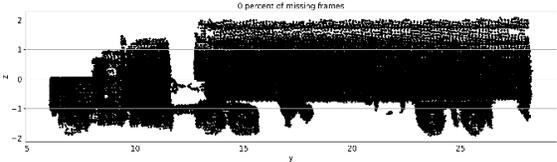
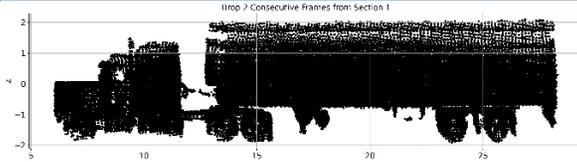
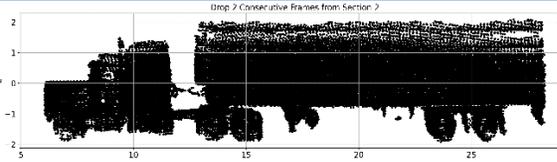
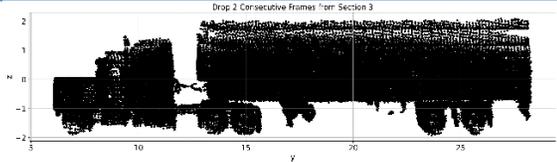
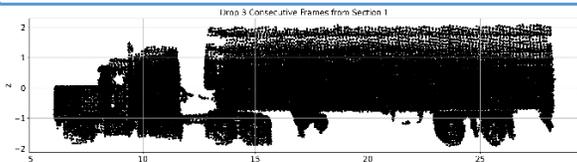
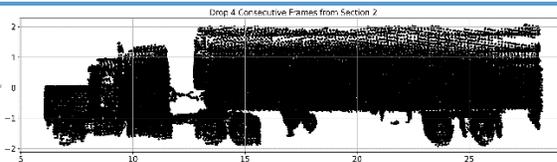
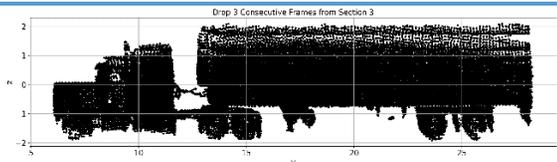
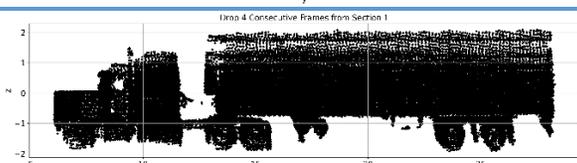
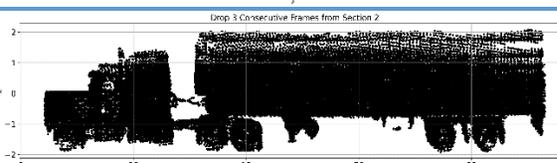
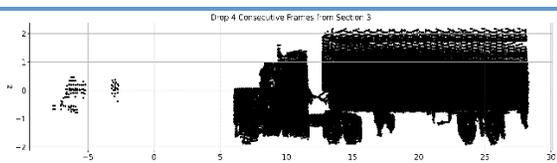
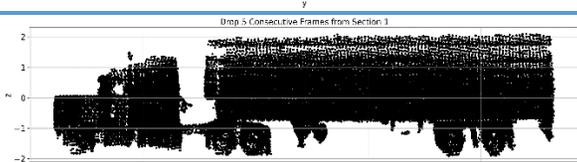
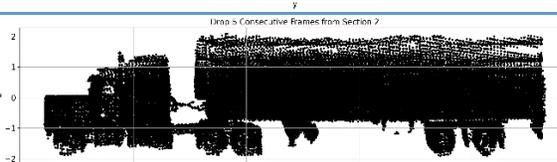
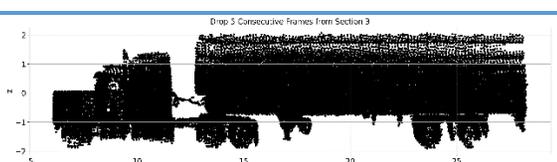
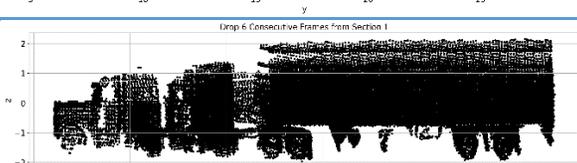
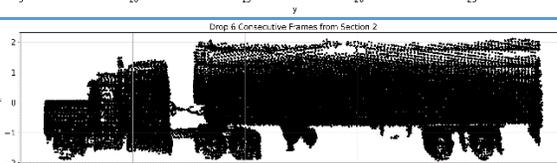
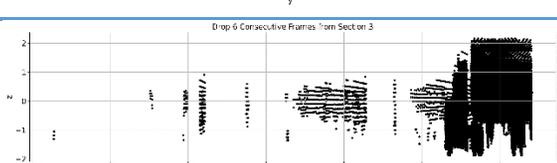
The data used in this study was collected from the single-lane off-ramp area. Therefore, the occluded vehicle point clouds were rarely observed at this data collection site. To test the robustness of the new registration framework, random frames for a truck object were dropped to simulate the missing frames scenario caused by vehicle occlusions. Figure 4.7 demonstrates the experiment of the missing frame test.



**Figure 4.7 Illustration of Experiment Design**

The duration that each vehicle traversing the LDZ was divided into three equal time slots denoted by Section 1, Section 2, and Section 3. Since vehicle occlusions generally happen on consecutive frames,  $m$  random consecutive frames were dropped from each section at each time. Table 4.2 presents the results of the experiment.

Table 4.2 Experiment Results

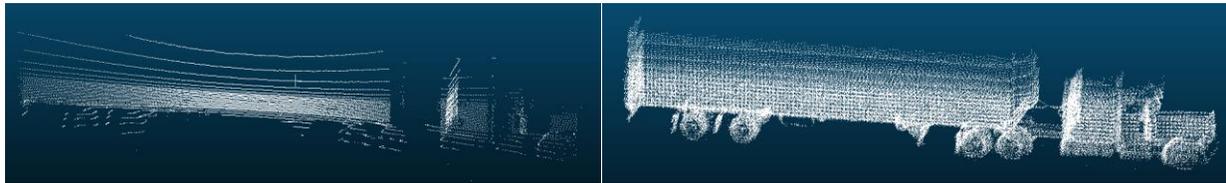
Number of Consecutive Frames	Section 1: Approaching the LiDAR Sensor	Section 2: In front of the LiDAR Sensor	Section 3: Leaving the LiDAR Sensor
0		0 percent of missing frames 	
2	Drop 2 Consecutive Frames from Section 1 	Drop 2 Consecutive Frames from Section 2 	Drop 2 Consecutive Frames from Section 3 
3	Drop 3 Consecutive Frames from Section 1 	Drop 4 Consecutive Frames from Section 2 	Drop 3 Consecutive Frames from Section 3 
4	Drop 4 Consecutive Frames from Section 1 	Drop 3 Consecutive Frames from Section 2 	Drop 4 Consecutive Frames from Section 3 
5	Drop 5 Consecutive Frames from Section 1 	Drop 5 Consecutive Frames from Section 2 	Drop 5 Consecutive Frames from Section 3 
6	Drop 6 Consecutive Frames from Section 1 	Drop 6 Consecutive Frames from Section 2 	Drop 6 Consecutive Frames from Section 3 
7	Completely miss aligned	Completely miss aligned	Completely miss aligned

When 5 consecutive frames are dropped, meaning that 0.5 seconds of data are missing, from either Section 1 or Section 2, the reconstructed point cloud is still able to preserve the essential information that can be used to identify their FHWA classes. For Section 3, the reconstruction framework fails when the 4 consecutive frames were dropped. The random 5 consecutive frames dropped from Section 3 were the last 5 frames used for the vehicle reconstruction. Therefore, nearly a quarter of the points on the rear truck wheel were missing.

This experiment demonstrated that the proposed framework is capable of reconstructing vehicle objects with 3-5 consecutive missing frames. A comprehensive vehicle occlusion analysis will be explored after real-world occlusion data are collected.

## 5. FHWA Axle-based Classification

The lower profile of a truck contained information related to its axle and general body configuration which defines their FHWA-CA classes. Compared to each frame of a truck object, the lower profile of the reconstructed truck point cloud is well-defined (Figure 5.1).

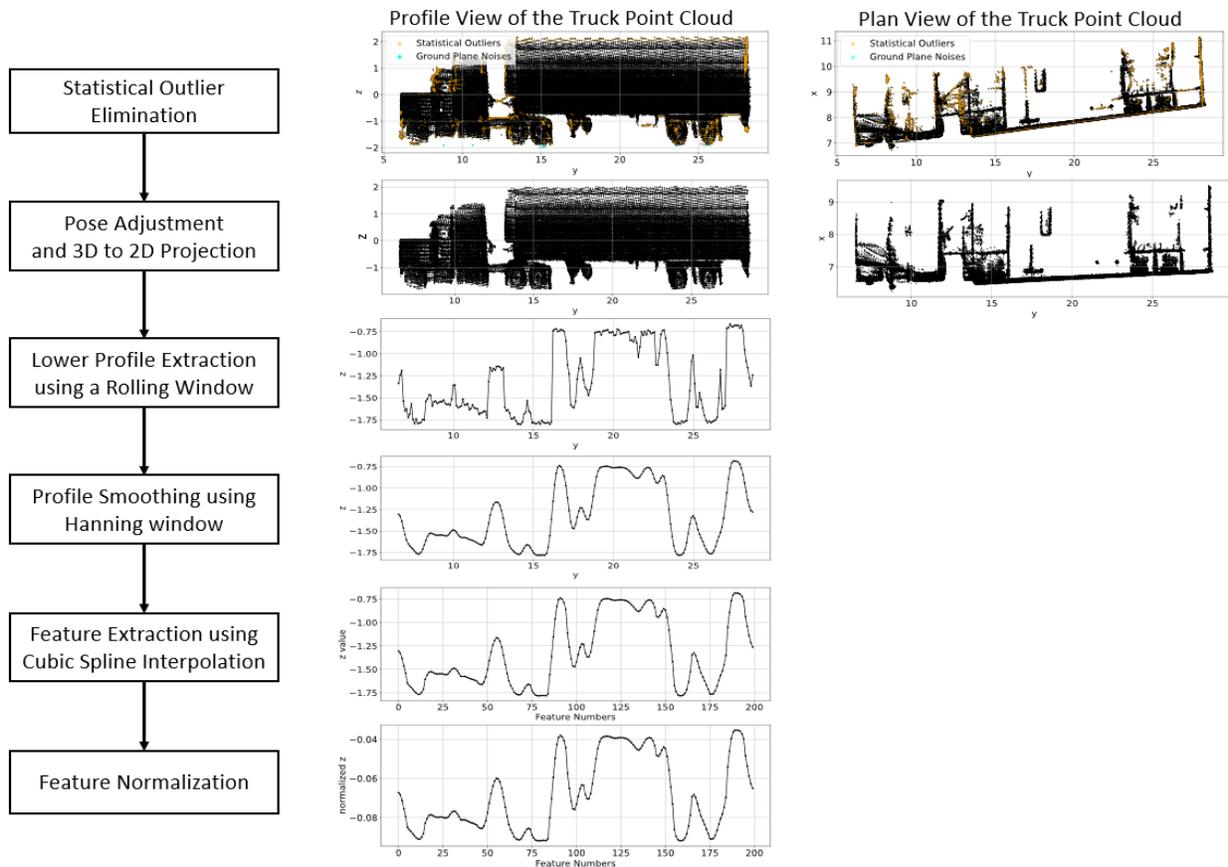


**Figure 5.1 A truck object represented by a sparse point cloud from a single frame (Left) vs. a dense point cloud representation reconstructed from multiple frames (Right).**

Therefore, in this section, essential features from the lower profile of the reconstructed truck point cloud were extracted and used as inputs for the vehicle classification model. Next, a deep ensemble neural network model was developed to assign vehicle point clouds to their corresponding FHWA-CA classes.

### 5.1 Feature Extraction

Statistical outliers on the reconstructed vehicle point cloud were further removed prior to the feature extraction (26). Subsequently, the pose of the vehicle point clouds was adjusted to align them with the  $zy$  plane using transformation matrix  $T_{mid-1,mid}$  since the middle frame was used as the reference frame in the pose graph optimization. Then, the 3D point cloud was projected to  $zy$  plane to obtain its 2D representation of each vehicle. The feature extraction process is shown Figure 5.2.



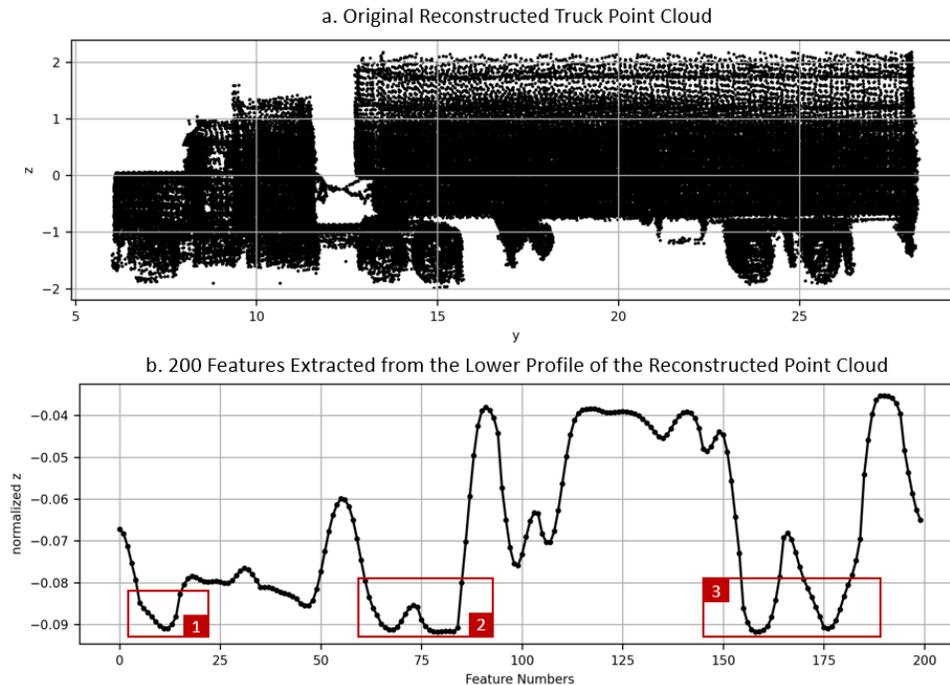
**Figure 5.2 Feature Extraction**

First, a rolling window with a size of 0.1 was created, where the minimum  $z$  value within the window was calculated. The size of the rolling window should be less than the radius of a regular wheel of a truck. The minimum  $z$  value rolling window captures the raw lower profile of each vehicle point cloud. Second, in order to obtain a better representation of the lower profile, the raw profile was smoothed using Hann window (29) which is formulated as:

$$w(i) = 0.5 - 0.5 \cos\left(\frac{2\pi i}{M-1}\right) \quad 0 \leq i \leq M-1 \quad (12)$$

Where  $i$  represents the index of each point in the profile.  $M$  is the window size of the filter.

The smoothed lower profile of the truck point cloud presents both the axle and general body configuration of the truck. Third, the smoothed lower profiles were interpolated using cubic spline interpolation, and then 200 equally spaced  $z$  values were extracted from the interpolated profile to align the dimension of the training instances. Finally, the interpolated profile was normalized between the limits of -1 to 1.

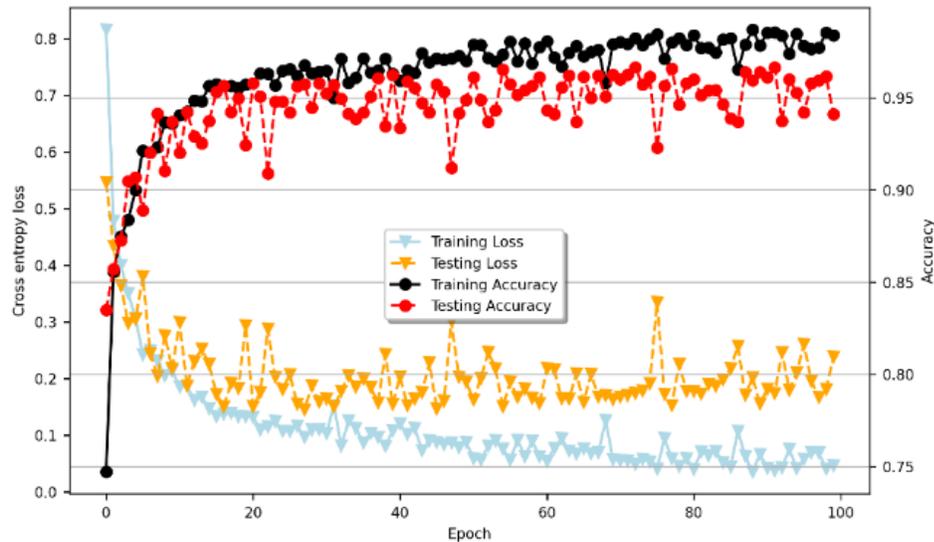


**Figure 5.3 Illustration of Features**

As Figure 5.3 shows, the valley in box 1 indicated the steering axle of the truck. Valleys in box 2 represent the drive axles of the tractor and the valleys in box 3 correspond the spread axles of the trailer. The peak shown in box 2 presents the connector between the tractor and trailer unit.

## 5.2 Bootstrap Aggregating Deep Neural Network for Vehicle Classification

Neural Network models have been proven to be able to approximate any complex non-linear mapping functions (30). Compared to a shallow neural network, the multi-layer structure of a deep neural network model allows it to accomplish the same task with exponentially lower computation complexity (31). Therefore, this study developed a deep neural network (DNN) with dropout regularization (32) to assign each vehicle point cloud to its corresponding FHWA-CA classes. The DNN model comprised 5 hidden layers with 512 neurons on each layer. Thirty percent of neurons were randomly dropped out on the last two hidden layers to avoid overfitting. The Rectified Linear Unit (ReLU) (33) with the He initialization method (34) was applied to each hidden layer and the Softmax activation function with Xavier initialization (35) was used on the output layer. The learning curve shown in Figure 5.4 traces the model performance histories during the training and testing process. After 100 epochs, the overall accuracy on the training set keeps gradually increasing while the testing accuracy converges to 0.95. Hence the model training converged after 100 epochs.



**Figure 5.4 Learning Curve**

In order to reduce the variability of the DNN prediction results, a bootstrap aggregating (bagging) (36) ensemble approach was applied. In this study, the bagging ensemble method resamples the training set with stratified bootstrap resampling strategy (37) to ten sets of bootstrapped training samples which were used to build ten different DNN models with the same model structure. The final prediction results were determined by the highest averaged prediction score of the ten DNN models.

### 5.3 Model Results

This section first presents the test results of the proposed model using a normalized confusion matrix and then provides the error analysis on the misclassified vehicles. In addition, the proposed model was compared with the state-of-the-art FHWA axle-based classification model using a LiDAR sensor.

#### 5.3.1 Classification Results and Analysis

The normalized confusion matrix of the classification model is presented in Figure 5.5.

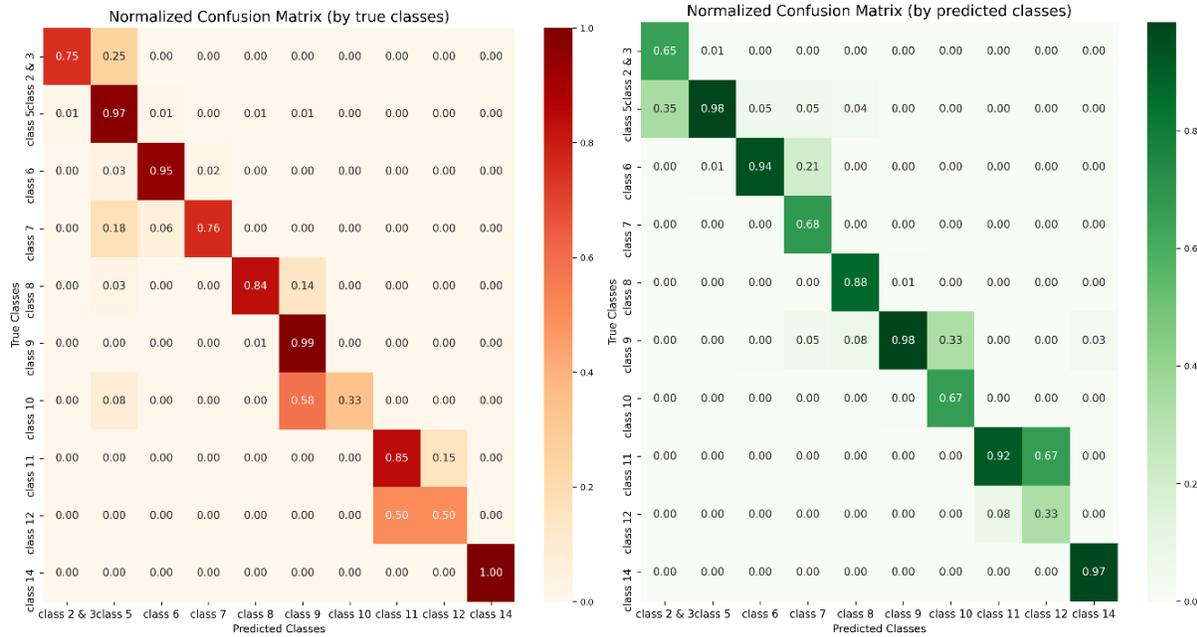
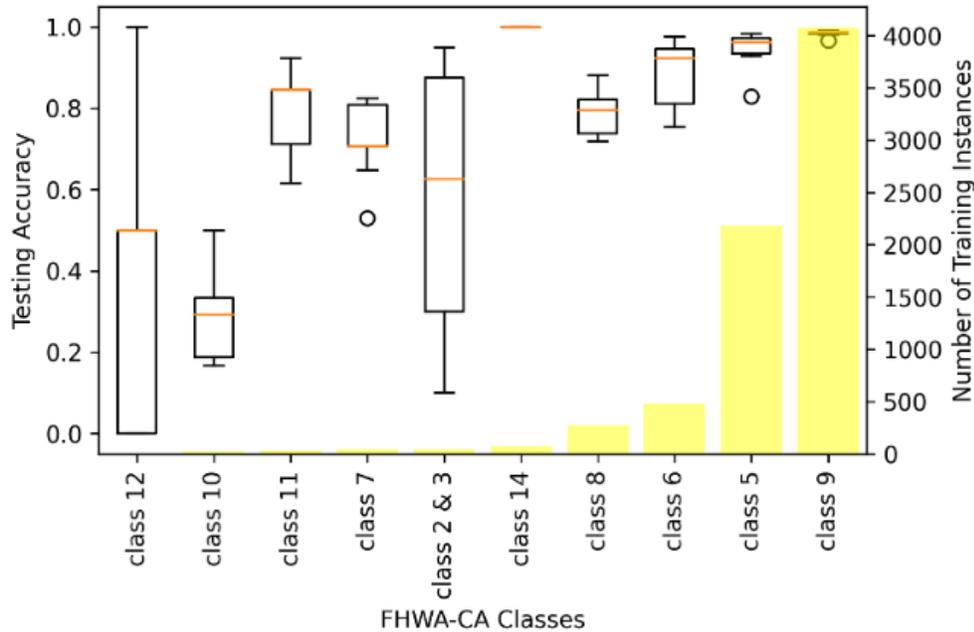


Figure 5.5 Normalized Confusion Matrix for the Test Set

Each row of the red-colored confusion matrix is normalized by a total number of groundtruth vehicles in their corresponding classes. Therefore, the diagonal elements represent the recall values of each class, which was also referred to as “Correct Classification Rate” (CCR) in some literature (8, 38). Each column of the green-colored confusion matrix is normalized according to the total number of predicted values for each class. Hence, the diagonal elements are the precision values of each class.

Based on the normalized confusion matrices, the proposed model is able to correctly classify Classes 5, 6, 8, 11, and 14 with over 80 percent CCR. However, the model is weak in predicting Class 10 and Class 12. Interestingly, the precision value in Class 10 is higher than the recall value. This implies that the implementation of the model is expected to yield very few predictions on Class 10, but most of them are expected to be correctly classified. This is quite ideal for the model implementation. Conversely, Class 12 has a higher recall than its precision value which would cause that the model to return more Class 12 predictions, with most of them misclassified from other classes.

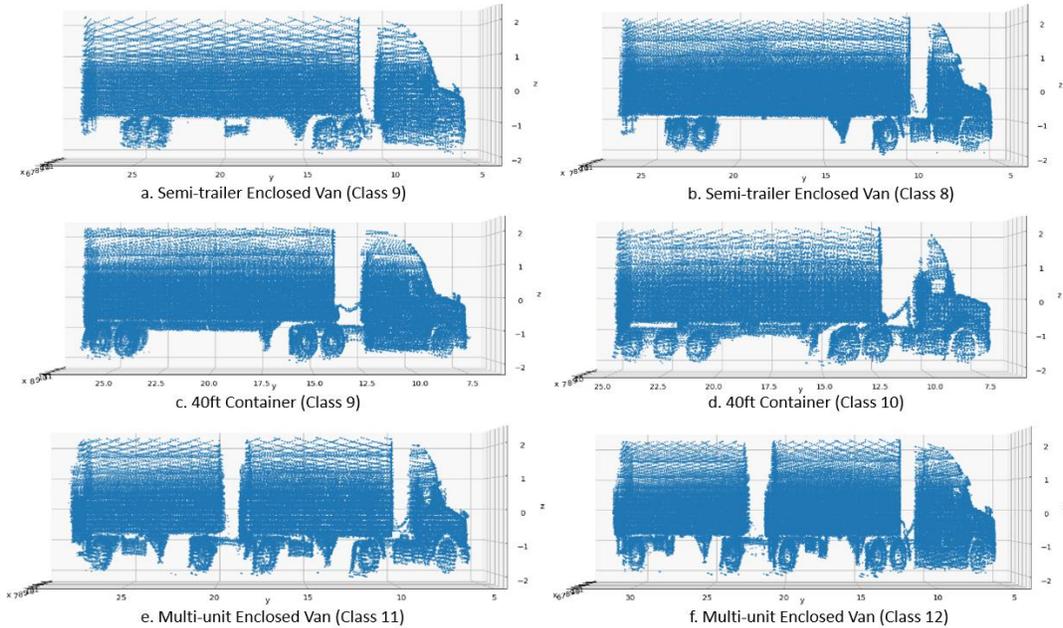
The boxplot in Figure 5.6 shows the model recall distribution of the DNN models which are built with 10 sets of bootstrapped training instances. The bar plot represents the training sample size for each class.



**Figure 5.6 CCR Distribution across All Classes**

As Figure 5.6 shows, the variability of the prediction results increases as the training sample size reduces, especially for Classes 10 and 12. Insufficient training samples were used to learn the key features from Classes 10 and 12 trucks which result in high variances in their prediction outcomes. In addition, Classes 2 and 3, passenger vehicles, are rarely observed at the entrance of the truck scale and those vehicles have larger diversity in terms of their body shape. Therefore, the model prediction variance is also high for Classes 2 and 3. Even though there is a limited number of training samples for Class 14, its prediction results are still promising since Class 14 represents a small homogeneous group of trucks in this dataset.

With sufficient training samples, the proposed classification model is capable of accurately distinguishing Classes 8 and 9 with overlapping body configuration (Figure 5.7a and b). However, classes with minor differences in their axle configuration but with the same body type are hard to distinguish when the training instances are not adequate (Figure 5.7c, d, e, and f). Consequently, in order to further enhance the model performance on Classes 10 and 12, the training dataset needs to be enriched in future studies.



**Figure 5.7 Overlapping Body Configurations**

### 5.3.2 Model Comparison

Table 5.1 provides a comparison between the model developed in this study with the state-of-the-art LiDAR-based classification model which used the single frame of an object to classify vehicles on the basis of the FHWA scheme (9).

**Table 5.1 Comparison between developed model (Bagging DNN) vs state of the art (Random Forest)**

FHWA-CA	CCR (Bagging DNN)	Testing Samples	Classes defined in (9)	CCR (Random Forest) (9)	Testing Samples (9)
Class 2	0.75	20	Passenger Vehicle	0.84	150
Class 3			Four-tire Single Unit	0.70	69
Class 4	None	None	Bus	1.00	20
Class 5 <sup>1</sup>	0.97	934	Two-axle, six-tire, single-unit truck	0.44	17
Class 6	0.95	208	Three-axle, single-unit truck	0.00	4
Class 7	0.76	17	Four or fewer axle, single-trailer truck	None	None
Class 8	0.84	117	None	None	None
Class 9 <sup>2</sup>	0.99	1,746	Five-axle, single-trailer truck	1.00	17
Class 10	0.33	12	None	None	None
Class 11	0.85	13	None	None	None
Class 12	0.50	2	None	None	None
Class 13	None	None	None	None	None
Class 14	1.00	31	None	None	None
<b>Average CCR</b>	0.79	-	-	0.76	-

Note: <sup>1</sup>Class 5 used in this study contained a two-axle truck pulling a small trailer which was not included in (9). <sup>2</sup>In the FHWA-CA scheme, Class 9 type 32 was separated from the rest of Class 9 truck and labeled as Class 14. In (9), Class 14 trucks are merged into Class 9 trucks.

Compared to the previous model (9), the new classification framework proposed in this study is able to classify vehicles in much more detail with significantly higher accuracy, especially for heavy-duty trucks from Class 8 to Class 14 which have disproportionately adverse impacts on pavement (39) and the environment (40).

## 6. Truck Body Type Classification

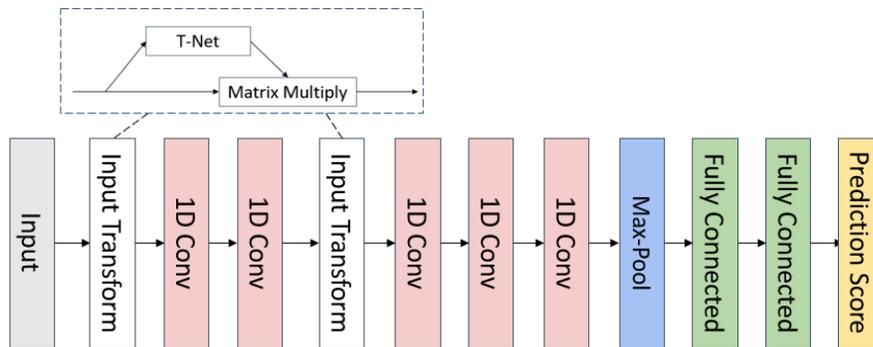
Transportation agencies have been increasingly interested in collecting truck body configuration data due to its strong association with industries and freight commodities, to better understand their distinct operational characteristics and impacts on infrastructure and the environment. In this section, this section explored the use of a novel deep neural network architecture - PointNet - to classify truck based on their body configurations.

### 6.1 PointNet-based Truck Classification Model

#### 6.1.1 The Deep Representation Learning Algorithm: PointNet

The reconstructed 3D point cloud is an irregular type of geometric data structure, where each point is represented by its cartesian coordinates  $(x, y, z)$ . A conventional convolution neural network requires a regular data format such as image pixels and 3D voxels as inputs. Therefore, the point cloud cannot be directly fed into a typical convolutional architecture. Point clouds are generally transformed to other data types for classification purposes. In the literature, transportation researchers usually extract high-level physical characteristics from either raw points (9) or transformed point clouds, e.g., 2D images (1–3, 7) or 3D voxels (8) to solve the truck classification problem. Finally, these high-level features are used

as inputs of classic machine learning algorithms. However, such data transformations and aggregations can introduce quantization error and further conceal the natural invariances of the point cloud data (41), which affects the accuracy and the variety of types of trucks that can be classified. In order to accommodate the characteristics of the point cloud data structure for improving truck classification accuracy, a novel deep neural network architecture—PointNet (41)—was adopted in this research. This neural network architecture can directly adopt point clouds as inputs and detect critical features for classification from the raw inputs. The architecture of PointNet is shown in Figure 6.1.



**Figure 6.1 PointNet Architecture (41)**

PointNet primarily benefits from two components of its architecture: the shared multi-layer perceptron (MLP) and the max-pooling function. The shared MLP was constructed using 1D convolution with a kernel size of 1, which provides a dense connection across points with the shared parameters (weight and bias terms). This means that the spatial encoding of each point can be learned by the shared MLP. A max-pooling layer was applied as a symmetric function to gather information from all the points, in order to resolve the invariance to permutation issue of the point cloud data structure. A function  $f(x_1, x_2, \dots, x_N)$  of  $N$  variables are invariant under random permutation if the function value does not change over the permutation of its variables. The generic representation of symmetric functions can be written as:

$$f(x_1, x_2, \dots, x_N) = f(x_N, x_2, \dots, x_1) = f(x_2, x_1, \dots, x_N) = \dots \quad (13)$$

The max-pooling function extracted the global critical feature of each truck point cloud and the overall model structure was able to learn the skeleton of each object. Since the truck body types are generally invariant and distinct in shape, PointNet ideally fits the task of truck body classification.

### 6.1.2 PointNet for Truck Classification Model

In this study, the PointNet architecture was adopted to classify truck body types in detail. Prior to the training process, the reconstructed point cloud needed to be regularized. First, the variable number of data points in reconstructed point clouds was uniformly downsampled to a common number of points as inputs into the PointNet. The downsampling process contained three steps. First, a regular voxel grid with a resolution of 5 percent was generated for each reconstructed truck point cloud, where those points were bucketed into voxels. Second, each occupied voxel was represented by a single point, which was calculated by taking the average of all points within each voxel grid. Finally, 1024 points suggested by Qi et al (41) were randomly sampled from the uniformly downsampled point cloud. A truck point

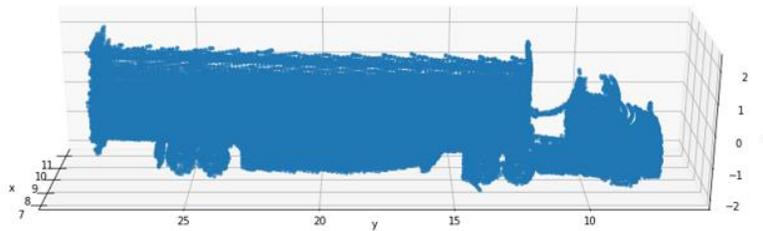
cloud  $k$  can be written as a 3D point set,  $p^k = \{(x_j^k, y_j^k, z_j^k) | j = 1, \dots, n\}$ , where  $n = 1024$  in this study. After the downsampling process, the centroid of the truck point cloud  $k$  was moved to the  $(0,0,0)$  point in the coordinate and was represented as  $p^{kc} = \{(x_j^{kc}, y_j^{kc}, z_j^{kc}) | j = 1, \dots, n\}$ . The operation along the  $x$  axis is presented in Equation 14, where  $y$  and  $z$  follow the same calculation.

$$x_j^{kc} = x_j^k - \frac{\max\{x_j^k\} - \min\{x_j^k\}}{2} \quad (14)$$

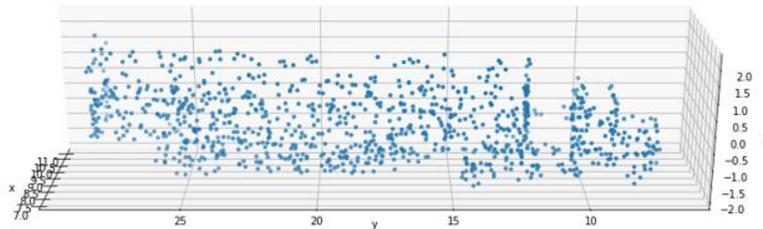
Then, the truck point cloud  $k$  was normalized to a unit sphere and denoted by  $p^{kcn} = \{(x_j^{kcs}, y_j^{kcs}, z_j^{kcs}) | j = 1, \dots, n\}$ . The operation along the  $x$ -axis is presented in Equation 15, where  $y$  and  $z$  follow the same calculation.

$$x_j^{kcs} = \frac{x_j^{kc}}{\max\{x_j^{kc}\}} \quad (15)$$

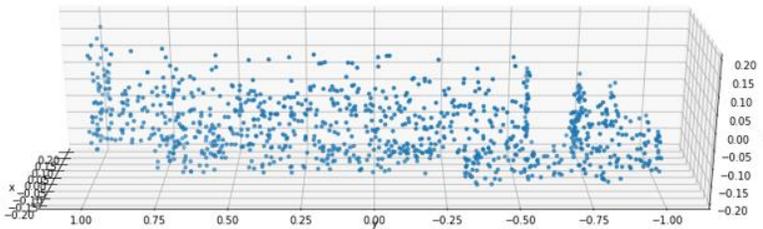
The point cloud preparation step is shown in Figure 6.2. Here, a reconstructed point cloud of an auto transport with a conventional tractor is taken as an example.



(a) Reconstructed Point Cloud of a Semi-trailer Enclosed Van



(b) Uniformly Downsampled Point Cloud (1024 Points)

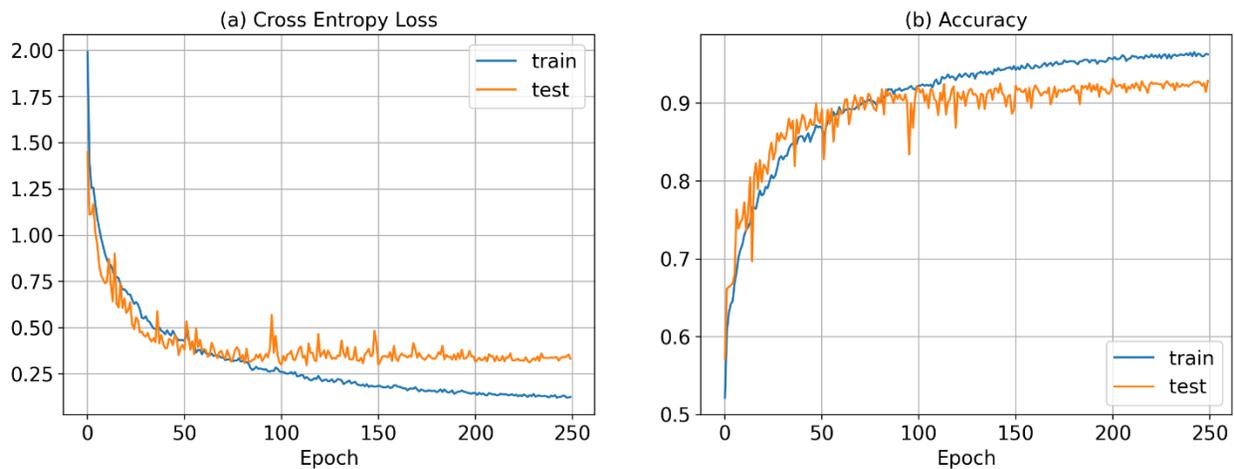


(c) Scaled Point Cloud

**Figure 6.2 Point Cloud Preprocessing**

During the model training process, two data augmentation methods were applied (41). First, each training instance was randomly rotated along the z-axis. Second, each point of the truck point cloud was jittered with a Gaussian noise which followed a  $N(0, 0.02)$  distribution to increase the diversity of the training instances.

The truck classification model was trained on 5,360 reconstructed truck point clouds with an RTX 2080 super GPU and took approximately 3 hours to converge. The learning process of the truck classification model is presented in the learning curve (Figure 6.3). The model accuracy on both training and test dataset improve in a similar trend until 100 epochs. After this point, the model performance gradually plateaus on the test dataset but continues improving on the training dataset. The model converged after 250 epochs.



**Figure 6.3 Learning Curves**

## 6.2 Model Averaging

A multiple layer structure with nonlinear activation functions on each layer provides deep neural networks with the ability to approximate any complex mapping function (42). However, deep neural network models generally suffer from high variance issues, where model performance varies significantly by dataset (43). Hence, model averaging strategies were explored to reduce the model variance and further enhance the model performance. The simplest way to apply model averaging on deep neural networks is to train multiple deep neural networks with different initial values and have all the models cast their votes. In this study, two model averaging methods were explored, and are explained in the next subsection.

### 6.2.1 Simple Model Averaging (SMA)

Let  $m_a = \{m_1, m_2, \dots, m_n\}$  denote  $n$  PointNet models trained with various initial values.  $c_b$  denotes the class labels.  $p(c_b|m_a)$  represents the probability that model  $m_a$  predicted class  $c_b$ . The equation of SMA is shown below.

$$\hat{c} = \underset{c_b \in \mathcal{C}}{\operatorname{argmax}} \sum_{a=1}^n p(c_b|m_a) \quad (16)$$

SMA assumes that  $m_a$  produced an equal contribution to the final decision and gave the prediction results by averaging all the votes of the candidate models.

### 6.2.2 Bayesian Model Averaging (BMA)

Unlike simple model averaging, which treats candidate models  $m_a$  equally, Bayesian model averaging assign a prior probability, presenting the subjective credibility of the model predicting a certain class. The posterior probability derived from the candidate models was used as the final prediction score (44, 45). In the case of a class  $c$  to be predicted based on training dataset  $D_{train}$  using  $n$  PointNets with initial value drawn from a normal distribution, the Bayesian model averaging provides final predictions based on the law of total probability:

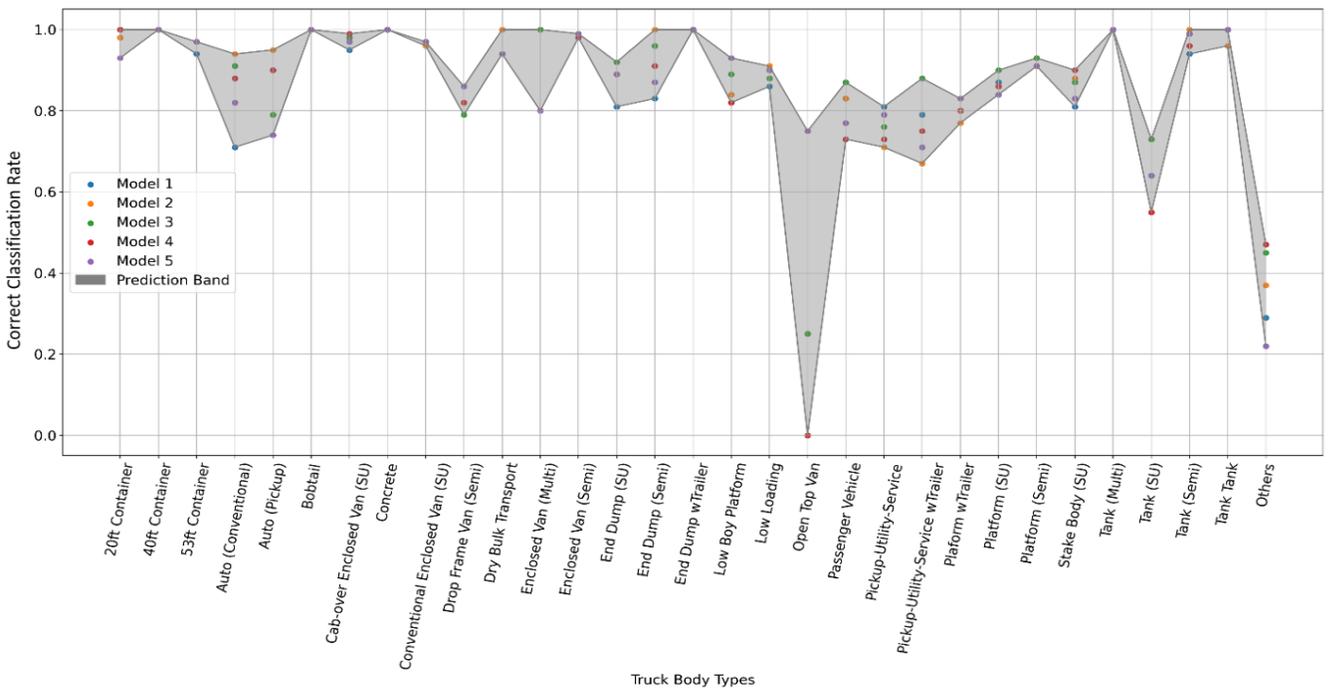
$$p(c) = \sum_{a=1}^N p(c_b|m_a)p(m_a|D_{train}) \tag{17}$$

$$\hat{c} = \underset{c_b \in C}{\operatorname{argmax}} \sum_{a=1}^N p(c_b|m_a)p(m_a|D_{train}) \tag{18}$$

As Equation 18 presents, the averaged model assigns higher weights to the candidate model which performs better for the specific class. The final prediction relies on the weighted average of the prediction scores.

### 6.3 Model Results

Five PointNet models were trained with different initial values. Figure 6.4 shows the CCR of each class from five different models.



## Figure 6.4 Prediction Variance Analysis

The same model structure yielded high variance to predict certain classes. For instance, model 4 obtained a CCR of 0 percent on predicting open-top vans, where model 5 was able to achieve a CCR of 75 percent on predicting the same class. Conversely, model 4 provided a CCR of 90 percent for single-unit stake body trucks. But the CCR for model 5 on predicting the same class was only 83 percent. Figure 6.4 thus reveals the need for an ensemble model. Auto (Conventional) and Auto (pickup) distinguish the tractor units of auto transport trucks. “Pickup-Utility-Service wTrailer” and “Platform wTrailer” are used to identify a straight driving unit pulling a small trailer. “End Dump wTrailer” considers an end dump truck pulling either a small trailer or another large dump trailer. “Other” represents all the truck types that do not fit the definition of the previous 30 classes. “Tank Tank” represents a tank tractor with a tank trailer, which specifically belongs to Class 14 in the California-modified FHWA scheme (46). The results from SMA and BMA are presented in Table 6.1.

**Table 6.1 Results of Body Classification Model**

	Model 1	Model 2	Model 3	Model 4	Model 5	SMA	BMA	Test Sample
20ft Container	0.98	0.98	1.00	1.00	0.93	1.00	1.00	59
40ft Container	1.00	1.00	1.00	1.00	1.00	0.99	0.99	196
53ft Container	0.94	0.97	0.97	0.97	0.97	0.96	0.96	170
Auto (Conventional)	0.71	0.94	0.91	0.88	0.82	0.91	0.91	34
Auto (Pickup)	0.90	0.95	0.79	0.90	0.74	0.89	0.89	19
Bobtail	1.00	1.00	1.00	1.00	1.00	1.00	1.00	109
Cab-over Enclosed Van (SU)	0.95	0.98	0.98	0.99	0.97	0.99	1.00	148
Concrete	1.00	1.00	1.00	1.00	1.00	1.00	1.00	16
Conventional Enclosed Van (SU)	0.97	0.96	0.97	0.97	0.97	0.97	0.97	362
Drop Frame Van (Semi)	0.82	0.82	0.79	0.82	0.86	0.82	0.79	28
Dry Bulk Transport	1.00	1.00	0.94	0.94	0.94	1.00	1.00	16
Enclosed Van (Multi)	0.80	0.80	1.00	0.80	0.80	0.80	0.80	5
Enclosed Van (Semi)	0.99	0.98	0.99	0.98	0.99	0.98	0.99	928
End Dump (SU)	0.81	0.89	0.92	0.89	0.89	0.88	0.88	26
End Dump (Semi)	0.83	1.00	0.96	0.91	0.87	0.96	0.96	23
End Dump wTrailer	1.00	1.00	1.00	1.00	1.00	1.00	1.00	7
Low Boy Platform	0.82	0.84	0.89	0.82	0.93	0.91	0.91	56
Low Loading	0.86	0.91	0.88	0.90	0.90	0.90	0.89	116
Open Top Van	0.00	0.25	0.25	0.00	0.75	0.25	0.00	4
Passenger Vehicle	0.83	0.83	0.87	0.73	0.77	0.80	0.80	30
Pickup-Utility-Service	0.81	0.71	0.76	0.73	0.79	0.79	0.79	94
Pickup-Utility-Service wTrailer	0.79	0.67	0.88	0.75	0.71	0.88	0.83	24
Platform wTrailer	0.80	0.77	0.80	0.80	0.83	0.83	0.83	30
Platform (SU)	0.87	0.86	0.90	0.86	0.84	0.90	0.89	135
Platform (Semi)	0.91	0.93	0.93	0.91	0.91	0.94	0.94	160
Stake Body (SU)	0.81	0.88	0.87	0.90	0.83	0.89	0.89	114
Tank (Multi)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	5
Tank (SU)	0.64	0.55	0.73	0.55	0.64	0.73	0.64	11
Tank (Semi)	0.94	1.00	0.99	0.96	0.99	0.99	0.99	83
Tank Tank	1.00	0.96	1.00	1.00	1.00	1.00	1.00	27
Others	0.29	0.37	0.45	0.47	0.22	0.37	0.18	49
Accuracy	0.92	0.93	0.94	0.93	0.93	0.94	0.94	3,084
Avg CCR	0.84	0.86	0.88	0.85	0.87	0.88	0.86	3,084

Note: SU: Single-Unit Truck; Semi: Tractors pulling Semi-Trailer; Multi: Tractors pulling multiple trailers. Cells labeled with red colors represent CCR lower than 0.80. Green colors highlight the benefits of using the SMA model.

After applying model averaging across five PointNet models, the number of classes with CCR value less than 80 percent was significantly reduced. The model ensemble outperformed most of the individual models in terms of accuracy, average class CCR, and F1 score. SMA and BMA presented the same level of accuracy according to these aggregated measurements. The SMA outperforms the BMA method in terms of model performance on drop frame vans (Semi), low loading truck, open-top vans, pickup/utility/service with trailer, and single-unit tank while the CCR of single-unit cab-over enclosed van was slightly reduced. The two-sided non-parametric Wilcoxon signed-rank test (47), was conducted

to test if a significant difference existed between the results of SMA and BMA. The p-value of 0.02 showed the null hypothesis for the difference between SMA and BMA was significant, and at a significance level of 5 percent could be rejected. Therefore, the performance of SMA is significantly better than the BMA method.

A closer assessment of the minority classes found that the performance of the ensemble PointNet was not significantly biased towards the majority class since the minority class presented low variations in their body type design. Hence, the ensemble PointNet presents a promising result in solving truck body type classification problems.

For illustration purposes, in Table 6.2 and Table 6.3, the “Others” class was split into “Others (SU)” and “Others (wTrailer)” to denote single-unit trucks and trucks with trailer(s) that were misclassified with other types of trucks respectively.

**Table 6.2 Confusion Matrix for single-unit truck and passenger vehicles (SMA Approach)**

	Bobtail	Cab-over Enclosed Van (SU)	Concrete Mixer	Conv Enclosed Van (SU)	End Dump (SU)	Low Loading	Passenger Vehicle	Pickup-Utility-Service	Platform (SU)	Stake Body (SU)	Tank (SU)	Others (SU)	Test Counts	CCR
Bobtail	109	0	0	0	0	0	0	0	0	0	0	0	109	1.00
Cab-over Enclosed Van (SU)	0	147	0	0	0	0	0	0	0	1	0	0	148	0.99
Concrete	0	0	16	0	0	0	0	0	0	0	0	0	16	1.00
Conv Enclosed Van (SU)	1	2	0	351	0	7	0	1	0	0	0	0	362	0.97
End Dump (SU)	0	0	0	0	23	0	0	0	1	2	0	0	26	0.88
Low Loading	0	1	0	7	0	104	0	4	0	0	0	0	116	0.90
Passenger Vehicle	0	0	0	0	0	1	24	5	0	0	0	0	30	0.80
Pickup-Utility-Service*	0	2	0	2	0	1	3	74	6	4	0	1	94	0.90
Platform (SU)	1	0	0	2	0	0	0	3	121	5	1	2	135	0.79
Stake Body (SU)**	0	0	0	2	1	0	0	3	4	102	0	1	114	0.89
Tank (SU)	0	0	0	0	0	0	0	0	0	3	8	0	11	0.73
Other (SU)	0	0	0	0	0	1	0	3	4	1	0	18	27	0.67

Note: Cells labeled with red colors represent CCR lower than 0.80. The yellow cells highlight the correctly classified numbers. The grey cells point to the main causes of the misclassification. “Conv” is short for “Conventional”. \*This row does not sum up to 94, since one of the pickup/utility/service trucks was misclassified with pickup/utility/service with a trailer. \*\* In this row, one stake body (SU) was misclassified as a semi-trailer platform. This was identified as mislabeling through visual verification.

Table 6.3 Confusion Matrix for a truck with Trailer(s) (SMA Approach)

	20ft Container	40ft Container	53ft Container	Auto (Conv)	Auto (Pickup)	Drop Frame (Semi)	Dry Bulk Transport	Enclosed Van (Multi)	Enclosed Van (Semi)	End Dump (Semi)	End Dump wTrailer	Low Boy Platform	Open Top Van	P/U/S wTrailer	Platform wTrailer	Platform (Semi)	Tank (Multi)	Tank (Semi)	Tank Tank	Others (wTrailer)	Test Counts	CCR	CCR in (8)
20ft Container	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	1.00	0.96
40ft Container	0	195	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	196	0.99	0.98
53ft Container	0	0	164	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	170	0.96	-
Auto (Conv)	0	0	0	31	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	34	0.91	0.91
Auto (Pickup)	0	0	0	1	17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	19	0.89	-
Drop Frame (Semi)	0	0	0	0	0	23	0	0	4	0	0	0	0	0	0	0	0	0	0	1	28	0.82	-
Dry Bulk Transport	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	16	1.00	-
Enclosed Van (Multi)	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	5	0.80	-
Enclosed Van (Semi)*	0	0	4	0	0	5	0	0	913	0	0	0	2	1	0	0	0	0	0	2	928	0.98	0.94
End Dump (Semi)	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	1	0	0	0	0	23	0.96	0.85
End Dump wTrailer	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	7	1.00	-
Low Boy Platform	0	0	0	0	0	0	0	0	0	0	0	51	0	1	0	4	0	0	0	0	56	0.91	-
Open Top Van	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	4	0.25	-
P/U/S wTrailer**	0	0	0	0	1	1	0	0	0	0	1	0	0	21	0	0	0	0	0	0	24	0.88	-
Platform wTrailer	1	0	0	0	0	0	0	0	0	0	1	0	0	2	25	0	0	0	0	1	30	0.83	-
Platform (Semi)	0	0	4	1	0	0	0	0	0	1	0	1	0	0	0	150	0	0	0	3	160	0.94	0.94
Tank (Multi)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	5	1.00	-
Tank (Semi)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	82	0	0	83	0.99	0.97
Tank Tank	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	27	1.00	-
Others (wTrailer)	0	0	7	0	1	1	0	2	4	0	0	0	2	0	1	2	1	1	0	18	40	0.45	-

Note: \* In this row, a semi-trailer enclosed van was misclassified to a pickup/utility/service truck. This was identified as mislabeling through visual verification. \*\* "P/U/S wTrailer" represents the pickup/utility/service truck

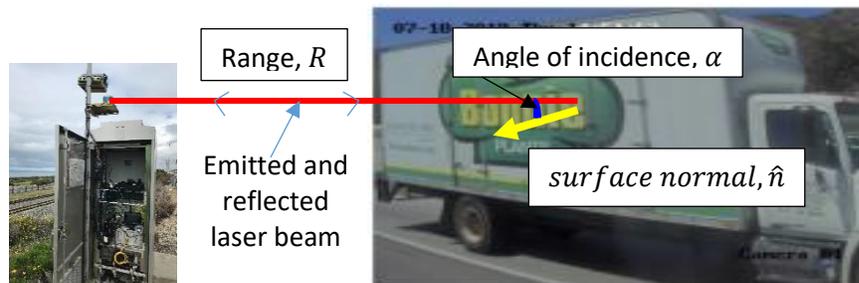
A comparison between the CCR values in Table 6.2 and Table 6.3 shows that the model was less competent in predicting single-unit trucks, where 18 percent of single-unit vehicles have an average CCR less than 0.80, while only 5 percent of trucks pulling trailer (s) have an average CCR less than 0.80. This was likely caused by the similarity across body types. For example, with different shapes of commodities or devices carried, single-unit platform trucks shared similar body configurations with pickup/utility/service trucks, single-unit stake body trucks, single-unit tank trucks, and single-unit dump trucks. In addition, the “passenger vehicle” class included 4-tire small pickups which shared a similar profile with 6-tire utility pickups that were categorized in the “pickup/utility/service” class.

Table 6.3 presents the confusion matrix of 19 truck body types, primarily including tractors pulling semi-trailers, tractors pulling a large single trailer, and tractors pulling multiple trailers. The body type confusion occurred primarily among auto transports, low boy platform, and semi-trailer platform trucks. Similar to the issues shown in single-unit trucks, the loading on the trailers is likely the cause of misclassifications across these three types. The performance of the SMA PointNet was compared with the state-of-art LiDAR-based classification model. SMA PointNet presented higher CCR values than the previous model across most classes, except semi-trailer platforms. In this sense, the ensemble PointNet can be considered superior to the state-of-the-art truck classification model (8). Platforms loaded with 53ft box containers were misclassified as 53ft box containers loaded on a container chassis, which was not included in the previous trailer type classification scheme (8). Classifying trucks in more detail naturally increases the chances of misclassification among similar body types. Therefore, balancing the total number of trucks that can be classified and high CCR values across all classes is critical.

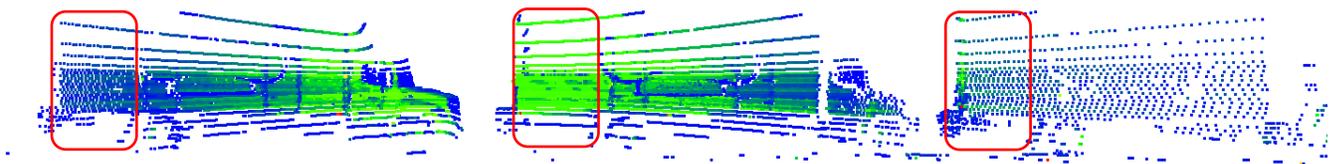
## 7. LiDAR intensity-based Truck Surface Characterization

### 7.1 Introduction

Aside from their physical attributes, fleet identification features such as logos found on many trucks can be used to infer their industry affiliation and can serve as another dimension of truck characterization to provide further insights into their activity patterns. Along with the depth and geometry information as mentioned in previous chapters, LiDAR sensors provide an additional attribute widely named as 'intensity'(I). LiDAR intensity is the measured power (returned) of a reflected laser beam from a target surface. These intensity values are indicator of surface reflectance of the target surface. For the Terrestrial LiDAR Scanner (TLS) in side-fire configuration, these intensity values are influenced by factors such as range ( $R$ ), angle of incidence ( $\alpha$ ), target surface roughness, and other instrument related variables. These factors are depicted in Figure 7.1. A significant advantage of LiDAR intensity over conventional images is its independence of ambient lighting, which allows it to be effective in poor lighting conditions, such as twilight and night times. This capability makes LiDAR a candidate technology for capturing truck fleet attributes.



**Figure 7.1 Factors affecting the LiDAR intensity**



**Figure 7.2 Illustration of need for Intensity homogenization**

The effect of above stated factors on LiDAR intensity can be seen in Figure 7.2. The three scans are colored by LiDAR intensity values ranging from 0-255 changing from blue shade for lower values to red shade for higher values. As a truck passes through the LiDAR Detection Zone (LDZ), the intensity values corresponding to the same part of truck change. This can be confirmed with the visual examination of Figure 7.2. For example, let us observe the intensity values corresponding to rear section of the truck body in the three scans. Intensity values change from blue shade to green and then back to blue shade.

If we would like to characterize the truck surfaces using LiDAR intensity values, it is necessary to homogenize the Intensity values such that same part of truck will have same intensity values irrespective of its position in LDZ. Homogenized LiDAR intensity values have potential and may enable consistent truck surface characterization even across multiple sites.

## 7.2 Proposed LiDAR Intensity Homogenization Framework

Radiometrically processed LiDAR intensity has been used in applications of Airborne LiDAR Scanning (ALS) for land cover estimation and classification, ecological monitoring, etc.(48). Terrestrial LiDAR Scanning (TLS) based intensity data has been used to investigate pavement markings (49–52), traffic sign reflectivity assessment (53), autonomous driving(48) etc. To the best of our knowledge, this is the first attempt of using LiDAR intensity data to characterize the surfaces of commercial vehicles. More comprehensive list of LiDAR intensity applications can be found in Table 1 of the reference (54).

### 7.2.1 Literature Review

LiDAR intensity can be expressed as the strength of backscattered laser echo from the scanned surface and is influenced by multiple complexly interacting factors. These factors include the geometry of the scanned surface, angle of incidence, range, environment, and the sensor itself. From literature, Range( $R$ ) and angle of incidence ( $\alpha$ ) are identified as the two most important variables that needs to be corrected for their influence on intensity (53). Majority of the intensity correction methods from literature involve lab-based calibration of empirical models involve estimation of parameters using standard targets of known intensity values. A comprehensive list of the empirical models from literature can be found in Table 4 of reference (54) and Table 2 of reference (48).

### 7.2.2 Proposed LiDAR intensity correction framework

Though empirical model driven Intensity correction methods can be accurate, they require calibration/estimation of parameters for every sensor in the lab preferably before the sensor is deployed for data collection. On the other hand, a data driven method for correction of intensity values could be applied to an existing / already collected data from the field. Hence our focus is to estimate the parameters for the LiDAR intensity correction using a data driven technique such as mentioned in (55).

As per the proposed framework the corrected intensity values( $I_c$ ) could be written as a function of observed intensity( $I_o$ ), range( $R$ ), and cosine of angle of incidence ( $\alpha$ )as mentioned in Equation 7-1 below.

$$I_c = I_o R^a \cos^b(\alpha)$$

#### Equation 7-1

To estimate the parameters in Equation 7-1, we adopt the assumption that a location on the of the truck body across several LiDAR scans of truck's trajectory should have same intensity value irrespective of its range and angle of incidence. Such points with their presence in more than one LiDAR scan are identified, their corresponding trajectory in successive scans is estimated. To make the parameter estimation process more robust, a very close neighboring region (patch) of each point is considered for the parameter estimation.

### 7.2.3 Calibration Strips for Intensity Correction

For simplifying the terminology in further documentation, let us call all the patches of one single point across scans as that point's calibration strip ( $cs_p$ ). So, each  $cs_p$  contains  $n_{cs,p}$  points with different range and angle of incidence. Technically all these points represent same region of truck, hence should have same intensity. Let us say we identified a total of  $N_{CS}$  such calibration strips for a given truck.

Then the parameter estimation problem could be stated as an optimization (minimization) problem which would parallelly reduce the variance of observed intensity values of each calibration strip of given truck. This optimization problem can be written as Equation 7-2.

$$\sum_{i_{csp} \in N_{CS}} \sum_{p \in i_{csp}} (\ln(I_{o,p}) + a \ln(R_p) + b \ln(\cos(\alpha_p)))^2$$

**Equation 7-2**

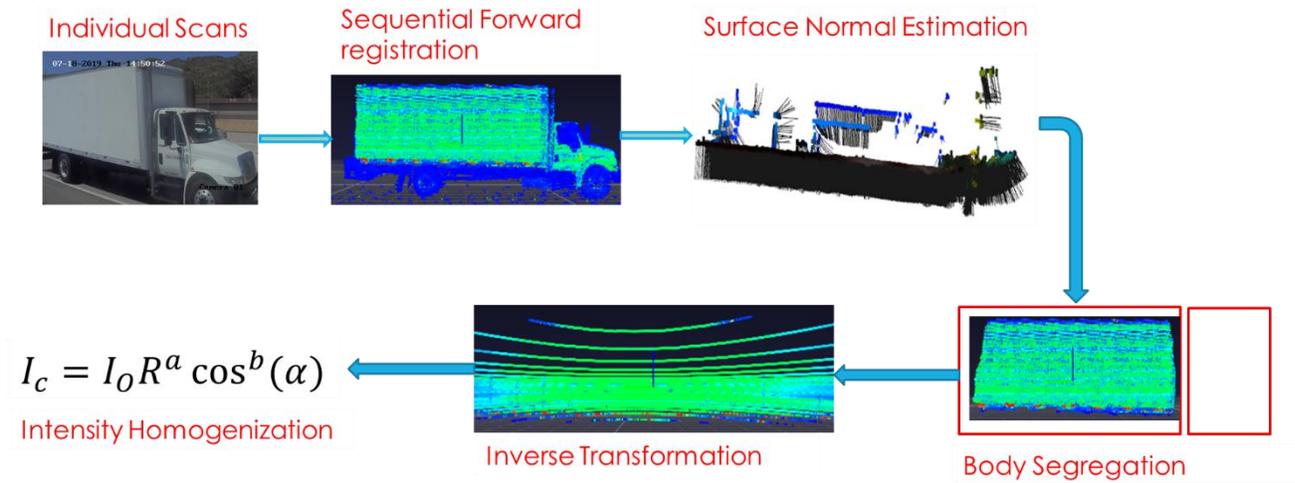
### 7.2.4 Stepwise Description of the Parameter Calibration

As explained in previous chapters each LiDAR scan of truck is a pointcloud object depicting the cartesian coordinates of truck geometry in 3D. Range and angle of incidence of each of those points needs to be calculated for estimating the Intensity correction parameters. The range of each of the points can be estimated directly by estimating the length of their position vector from the LiDAR scanner itself.

The surface normal of each of the points along with their position vectors is needed to estimate the angle of incidence. As discussed in section 3.5.1, the density of the points is not constant for each LiDAR scan both in horizontal as well as vertical direction.

The sparse point density of individual frames affects the ability to estimate accurate surface normal near the edges. This limitation was overcome by estimating the surface normals of the dense reconstructed truck in translated coordinates using the transformation matrices obtained from section 4.2. Once the surface normals estimates were obtained, the body of the truck was segregated by using a hybrid sequential Gaussian Mixture Models -based clustering of normals and density-based DBSCAN clustering. The surface normals of the segregated body of truck were inverse transformed back to the original coordinates for the purpose of angle of incidence estimation.

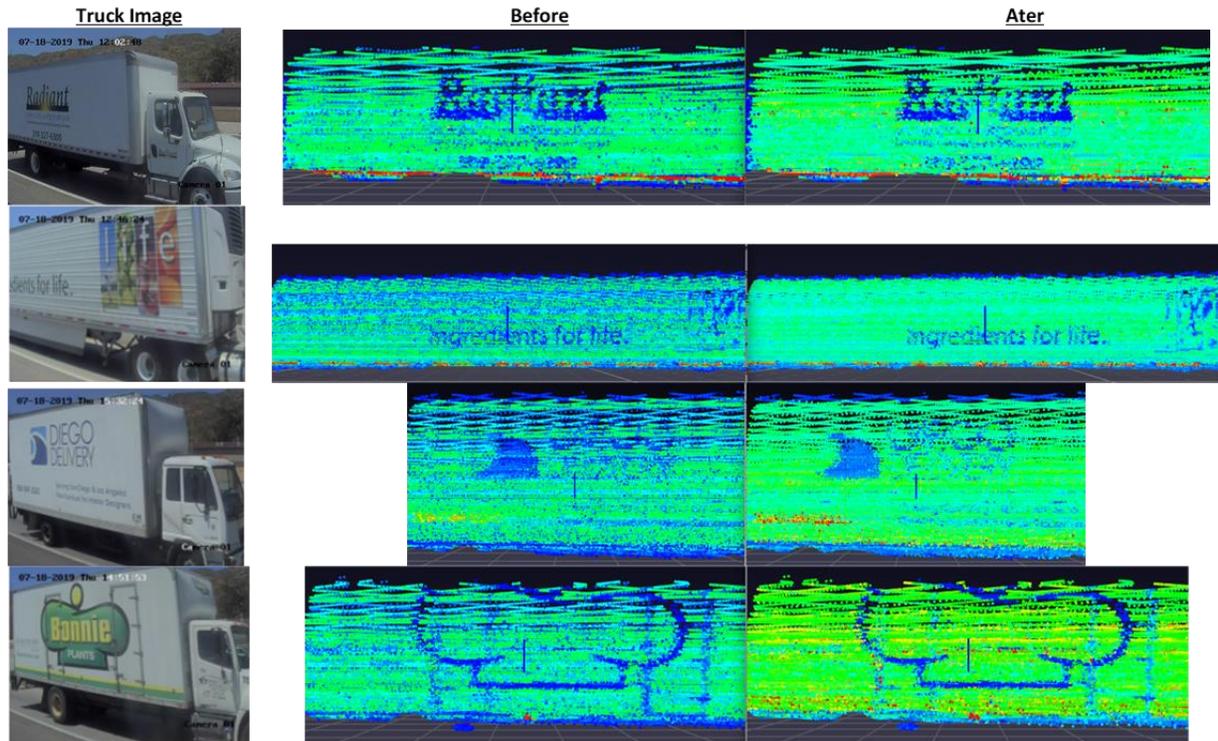
Once the surface normals are obtained, the calibration strips of the truck are obtained as described in section 7.2.3. An evolutionary algorithm based many objective optimization using pymoo (56) is performed on the calibration strips of a truck to estimate the Intensity correction parameters. This framework is illustrated in a flowchart in Figure 7.3.



**Figure 7.3 LiDAR Intensity correction framework**

### 7.3 Preliminary Results of LiDAR Intensity Homogenization

The proposed framework was applied to a small subset of trucks from section 3.3. Preliminary results presented in Figure 7.4 show an improved distinction of fleet identification features. The corrected intensity values show a reduction in the variance of intensity values quite well and provides a distinct contrast of fleet features against the background.



**Figure 7.4 Preliminary results of LiDAR Intensity correction**

### 7.4 Future Expansion

This LiDAR intensity correction framework has the potential to facilitate the characterization of trucks at fleet level. This could fill significant freight data gap and help provide valuable insights to freight policy making agencies.



**Figure 7.5 Potential to fill freight data gaps**

Two hypothetical example cases are provided to demonstrate the potential of a LiDAR intensity-based truck surface characterization model. Traditionally trucks 1.a and 1.b from Figure 7.5 would have been identified as FHWA Class 9 Semi tank truck by existing advanced truck classification methods(38). With the development of LiDAR intensity-based surface characterization, a new dimension of industry being served could be added. In this example truck 1.a mostly serves cryogenic industry, whereas truck 1.b serves the retail distribution centers of gasoline. Similarly for truck 2.a, 2.b we would have been able to identify a freight vs non-freight vehicle for the same body configuration with the availability of surface characterization model.

## 8. Conclusion

To fill the truck monitoring gaps on rural highway corridors, this study developed novel LiDAR-based truck classification methods through the development of a new truck point cloud reconstruction framework that was able to retain a wide LDZ and accurately classify trucks based on the FHWA-CA scheme and detailed truck body configurations. The data used for modeling was collected from a horizontally oriented multi-array 3D LiDAR sensor, which has the ability to capture a wide field of view of the roadway. In this case, even though vehicles traveling in the outermost lanes presented in front of the LiDAR sensor for a short period of time and occluded vehicles traveling in the corresponding inner lanes, the point cloud originating from those occluded vehicles could be retrieved from consecutive frames. The sparse point clouds from individual frames resulting from a low vertical resolution were enriched by aggregating multiple frames associated with the same truck. Subsequently, the lower profile of the reconstructed vehicle point cloud was extracted and used as inputs for the deep neural network to classify vehicles based on the FHWA classification scheme. The classification model with the reconstruction framework outperformed the state-of-the-art axle-based classification model using LiDAR sensors in terms of both their accuracy and robustness. This LiDAR-based FHWA model achieved a 79 percent average CCR. Classes 8 and 9 were classified correctly with 84 percent and 99 percent CCR even though they share very similar body configurations.

This study investigated the PointNet deep representation learning algorithm to further classify trucks in their detailed body configurations. The PointNet-based model successfully learned the basic characteristics of each truck class by selecting the critical features from each preprocessed point cloud. Finally, two model ensemble strategies, SMA and BMA, were explored to improve the generality of the model and to further enhance the model performance. The LiDAR-based truck body type classification model was able to classify heavy-duty trucks in much more detail, with a close relationship to their industry affiliations. For example, the new model could accurately distinguish low boy platforms from general flatbed trucks, where these two types of platform trucks are designed to carry different types of payloads. This model was able to classify 31 different vehicle types (advantageously mainly trucks) and achieve an average class CCR of 90 percent for both a truck with trailer (s) and single-unit vehicles. Remarkably, the proposed method was able to distinguish 53ft containers and semi-trailer enclosed vans with over 95 percent CCR even though they share very similar physical characteristics, which is a significant improvement over previous models using the integration of WIM and inductive signature data (38), as well as LiDAR (8).

In the future, more LiDAR data will be collected from other detection sites to test the transferability of the proposed model. Furthermore, multi-lane truck classification applications can be explored as the horizontal orientation of the LiDAR permits capturing a full 360-degree field of view.

## References

1. Abdelbaki, H. M., K. Hussain, and E. Gelenbe. A Laser Intensity Image Based Automatic Vehicle Classification System. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2001, pp. 460–465. <https://doi.org/10.1109/itsc.2001.948701>.
2. Hussain, K. F., and G. S. Moussa. Laser Intensity Vehicle Classification System Based on Random Neural Network. *Proceedings of the Annual Southeast Conference*, Vol. 1, 2005, pp. 131–135. <https://doi.org/10.1145/1167350.1167372>.
3. Sandhawalia, H., J. A. Rodriguez-Serrano, H. Poirier, and G. Csurka. Vehicle Type Classification from Laser Scanner Profiles: A Benchmark of Feature Descriptors. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, No. Itsc, 2013, pp. 517–522. <https://doi.org/10.1109/ITSC.2013.6728283>.
4. Lee, H., and B. Coifman. Side-Fire Lidar-Based Vehicle Classification. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2308, No. 1, 2013, pp. 173–183. <https://doi.org/10.3141/2308-19>.
5. Yang, R. Vehicle Detection and Classification from a LIDAR Equipped Probe Vehicle. 2009.
6. Asborn, M. I., C. G. Burris, and S. Hernandez. Truck Body-Type Classification Using Single-Beam Lidar Sensors. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2673, No. 1, 2019, pp. 26–40. <https://doi.org/10.1177/0361198118821847>.
7. Vatani Nezafat, R., O. Sahin, and M. Cetin. Transfer Learning Using Deep Neural Networks for Classification of Truck Body Types Based on Side-Fire Lidar Data. *Journal of Big Data Analytics in Transportation*, Vol. 1, No. 1, 2019, pp. 71–82. <https://doi.org/10.1007/s42421-019-00005-9>.
8. Sahin, O., R. V. Nezafat, and M. Cetin. Methods for Classification of Truck Trailers Using Side-Fire Light Detection and Ranging (LiDAR) Data. *Journal of Intelligent Transportation Systems*, 2020, pp. 1–13. <https://doi.org/10.1080/15472450.2020.1733999>.
9. Wu, J., H. Xu, Y. Zheng, Y. Zhang, B. Lv, and Z. Tian. Automatic Vehicle Classification Using Roadside LiDAR Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2019, p. 036119811984385. <https://doi.org/10.1177/0361198119843857>.
10. *VLP-32C User Manual*. 2018.
11. Hernandez, S. V. *Integration of Weigh-In-Motion and Inductive Signature Data for Truck Body Classification*. 2014.
12. Daszykowski, M., and B. Walczak. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Kdd*, Vol. 96, 1996, pp. 226–231. <https://doi.org/10.1016/B978-044452701-1.00067-3>.
13. Kriegel, H.-P., P. Kröger, K. Kröger, J. " Org Sander, and A. Zimek. Density-Based Clustering. <https://doi.org/10.1002/widm.30>.
14. Bewley, A., Z. Ge, L. Ott, F. Ramos, and B. Uproft. Simple Online and Realtime Tracking. No. 2016-Augus, 2016, pp. 3464–3468.
15. Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Fluids*

- Engineering, Transactions of the ASME*, Vol. 82, No. 1, 1960, pp. 35–45.  
<https://doi.org/10.1115/1.3662552>.
16. Kuhn, H. W. The Hungarian Method for the Assignment Problem. *Naval Research Logistics*, Vol. 2, No. 1–2, 1955, pp. 83–97. <https://doi.org/10.1002/nav.20053>.
  17. Besl, P., and N. McKay. A Method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239–256.
  18. Chen, Y., and G. Medioni. Object Modeling by Registration of Multiple Range Images. *Proceedings - IEEE International Conference on Robotics and Automation*. 2724–2729.
  19. Pomerleau, F., F. Colas, and R. Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, Vol. 4, No. 1, 2015, pp. 1–104.  
<https://doi.org/10.1561/23000000035i>.
  20. Pomerleau, F., F. Colas, R. Siegwart, S. Magnenat, F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, C. Icp, and F. C. Roland. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, Vol. 34, No. 3, 2013, pp. 133–148.
  21. Myronenko, A., and X. Song. Point Set Registration: Coherent Point Drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 12, 2010, pp. 2262–2275.  
<https://doi.org/10.1109/TPAMI.2010.46>.
  22. Jian, B., I. C. Society, and B. C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 8, 2011, pp. 1633–1645.
  23. Horaud, R., F. Forbes, M. Yguel, G. Dewaele, and J. Zhang. Rigid and Articulated Point Registration with Expectation Conditional Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 3, 2011, pp. 587–602. <https://doi.org/10.1109/TPAMI.2010.94>.
  24. Evangelidis, G. D., D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A Generative Model for the Joint Registration of Multiple Point Sets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 8695 LNCS, No. PART 7, 2014, pp. 109–122. [https://doi.org/10.1007/978-3-319-10584-0\\_8](https://doi.org/10.1007/978-3-319-10584-0_8).
  25. Gao, W., and R. Tedrake. Filterreg: Robust and Efficient Probabilistic Point-Set Registration Using Gaussian Filter and Twist Parameterization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2019-June, 2019, pp. 11087–11096.  
<https://doi.org/10.1109/CVPR.2019.01135>.
  26. Zhou, Q.-Y., J. Park, and V. Koltun. *Open3D: A Modern Library for 3D Data Processing*.
  27. Choi, S., Q. Y. Zhou, and V. Koltun. Robust Reconstruction of Indoor Scenes. No. 07-12-June, 2015, pp. 5556–5565.
  28. Allu, K., Z. Sun, and A. Tok. LiDAR-Based Reconstruction for Truck Surveillance. *UC Irvine: Institute of Transportation Studies*, 2020.
  29. Oppenheim, V., A., R. Schafer, W., and J. Buck, R. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 1999.

30. Hornik, K., M. Stinchcombe, and H. White. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
31. Shiyu Liang; R. Srikant. Why Deep Neural Networks for Function Approximation? 2017.
32. Mele, B., and G. Altarelli. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Physics Letters B*, Vol. 299, No. 3–4, 2014, pp. 345–350. [https://doi.org/10.1016/0370-2693\(93\)90272-J](https://doi.org/10.1016/0370-2693(93)90272-J).
33. Nair, V., and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, No. 3, 2010, pp. 807–814. <https://doi.org/10.1.1.165.6419>.
34. He, K., X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2015 Inter, 2015, pp. 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>.
35. Glorot, X., and Y. Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Journal of Machine Learning Research*, Vol. 9, 2010, pp. 249–256.
36. Breiman, L. Bagging Predictors: Technical Report No. 421. *Department of Statistics University of California*, No. 2, 1994, p. 19.
37. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, Vol. 2, No. 5, 1988, pp. 347–370.
38. Hernandez, S. V., A. Tok, and S. G. Ritchie. Integration of Weigh-in-Motion (WIM) and Inductive Signature Data for Truck Body Classification. *Transportation Research Part C: Emerging Technologies*, Vol. 68, 2016, pp. 1–21. <https://doi.org/10.1016/j.trc.2016.03.003>.
39. Gillespie, T. D., Karamihas, S. M. & Sayer, M. W. *Effects of Heavy-Vehicle Characteristics on Pavement Response and Performance (NCHRP Report 353)*. 1993.
40. Guensler, R., S. Yoon, H. Li, and J. Jun. *Heavy-Duty Diesel Vehicle Modal Emission Model (HDDV-MEM) Volume I: Modal Emission Modeling Framework*. 2005.
41. Qi, C. R., H. Su, K. Mo, and L. J. Guibas. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*.
42. Goodfellow, I., Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
43. Zhou, Z. H., J. Wu, and W. Tang. Ensembling Neural Networks: Many Could Be Better than All. *Artificial Intelligence*, Vol. 137, No. 1–2, 2002, pp. 239–263. [https://doi.org/10.1016/S0004-3702\(02\)00190-X](https://doi.org/10.1016/S0004-3702(02)00190-X).
44. Hoeting, J. A., D. Madigan, A. E. Raftery, C. T. Volinsky, A. E. Raftery, and C. T. Volinsky. Bayesian Model Averaging: A Tutorial (with Discussion). *Statistical Science*, Vol. 14, No. 4, 1999, pp. 382–417.
45. Raftery, A. E., T. Gneiting, F. Balabdaoui, and M. Polakowski. Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review*, Vol. 133, No. 5, 2005, pp. 1155–1174. <https://doi.org/10.1175/MWR2906.1>.

46. Quinley, R. WIM Data Analyst's Manual: FHWA Report IF-10-018. 2010, p. 183.
47. Wilcoxon, F. Individual Comparisons of Grouped Data by Ranking Methods. *Journal of economic entomology*, Vol. 39, No. 6, 1946, p. 269. <https://doi.org/10.1093/jee/39.2.269>.
48. Sanchiz-Viel, N., E. Bretagne, E. M. Mouaddib, and P. Dassonville. Radiometric Correction of Laser Scanning Intensity Data Applied for Terrestrial Laser Scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 172, 2021, pp. 1–16. <https://doi.org/10.1016/J.ISPRSJPRS.2020.11.015>.
49. Cheng, Y. T., A. Patel, C. Wen, D. Bullock, and A. Habib. Intensity Thresholding and Deep Learning Based Lane Marking Extraction and Lanewidth Estimation from Mobile Light Detection and Ranging (LiDAR) Point Clouds. *Remote Sensing*, Vol. 12, No. 9, 2020, p. 1379. <https://doi.org/10.3390/RS12091379>.
50. Gargoum, S., and K. El-Basyouny. Automated Extraction of Road Features Using LiDAR Data: A Review of LiDAR Applications in Transportation. 2017.
51. Hata, A., and D. Wolf. Road Marking Detection Using LIDAR Reflective Intensity Data and Its Application to Vehicle Localization. 2014.
52. Jung, J., E. Che, M. J. Olsen, and C. Parrish. Efficient and Robust Lane Marking Extraction from Mobile Lidar Point Clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 147, No. June 2018, 2019, pp. 1–18. <https://doi.org/10.1016/j.isprsjprs.2018.11.012>.
53. Ai, C., and Y. J. Tsai. An Automated Sign Retroreflectivity Condition Evaluation Methodology Using Mobile LIDAR and Computer Vision. *Transportation Research Part C: Emerging Technologies*, Vol. 63, 2016, pp. 96–113. <https://doi.org/10.1016/j.trc.2015.12.002>.
54. Kashani, A. G., M. J. Olsen, C. E. Parrish, and N. Wilson. A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration. *Sensors (Switzerland)*. 11. Volume 15, 28099–28128. [/pmc/articles/PMC4701271/?report=abstract](https://doi.org/10.3390/s150928099). Accessed Sep. 23, 2020.
55. Jutzi, B., and H. Gross. NORMALIZATION OF LIDAR INTENSITY DATA BASED ON RANGE AND SURFACE INCIDENCE ANGLE. 2009.
56. Blank, J., and K. Deb. Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, Vol. 8, 2020, pp. 89497–89509. <https://doi.org/10.1109/ACCESS.2020.2990567>.