

Old Dominion University

ODU Digital Commons

Civil & Environmental Engineering Theses & Dissertations

Civil & Environmental Engineering

Spring 2020

Truck Trailer Classification Using Side-Fire Light Detection And Ranging (LiDAR) Data

Olcay Sahin

Old Dominion University, olcaysah@gmail.com

Follow this and additional works at: https://digitalcommons.odu.edu/cee_etds



Part of the [Artificial Intelligence and Robotics Commons](#), [Transportation Commons](#), and the [Transportation Engineering Commons](#)

Recommended Citation

Sahin, Olcay. "Truck Trailer Classification Using Side-Fire Light Detection And Ranging (LiDAR) Data" (2020). Doctor of Philosophy (PhD), Dissertation, Civil & Environmental Engineering, Old Dominion University, DOI: 10.25777/ca1j-8628
https://digitalcommons.odu.edu/cee_etds/112

This Dissertation is brought to you for free and open access by the Civil & Environmental Engineering at ODU Digital Commons. It has been accepted for inclusion in Civil & Environmental Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**TRUCK TRAILER CLASSIFICATION USING SIDE-FIRE LIGHT DETECTION
AND RANGING (LiDAR) DATA**

by

Olcay Sahin

B.S. May 2005, International Black Sea University – Tbilisi, Georgia

M.B.A. May 2008, Strayer University

M.Sc. December 2012, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

CIVIL AND ENVIRONMENTAL ENGINEERING

OLD DOMINION UNIVERSITY

May 2020

Approved by:

Mecit Cetin (Director)

Sherif Ishak (Member)

Hong Yang (Member)

ABSTRACT

TRUCK TRAILER CLASSIFICATION USING SIDE-FIRE LIGHT DETECTION AND RANGING (LiDAR) DATA

Olcay Sahin
Old Dominion University, 2020
Director: Dr. Mecit Cetin

Classification of vehicles into distinct groups is critical for many applications, including freight and commodity flow modeling, pavement management and design, tolling, air quality monitoring, and intelligent transportation systems. The Federal Highway Administration (FHWA) developed a standardized 13-category vehicle classification ruleset, which meets the needs of many traffic data user applications. However, some applications need high-resolution data for modeling and analysis. For example, the type of commodity being carried must be known in the freight modeling framework. Unfortunately, this information is not available at the state or metropolitan level, or it is expensive to obtain from current resources.

Nevertheless, using current emerging technologies such as Light Detection and Ranging (LiDAR) data, it may be possible to predict commodity type from truck body types or trailers. For example, refrigerated trailers are commonly used to transport perishable produce and meat products, tank trailers are for fuel and other liquid products, and specialized trailers carry livestock. The main goal of this research is to develop methods using side-fired LiDAR data to distinguish between specific types of truck trailers beyond what is generally possible with traditional vehicle classification sensors (e.g., piezoelectric sensors and inductive loop detectors).

A multi-array LiDAR sensor enables the construction of 3D-profiles of vehicles since it measures the distance to the object reflecting its emitted light. In this research 16-beam LiDAR sensor data are processed to estimate vehicle speed and extract useful information and features to

classify semi-trailer trucks hauling ten different types of trailers: a reefer and non-reefer dry van, 20 ft and 40 ft intermodal containers, a 40 ft reefer intermodal container, platforms, tanks, car transporters, open-top van/dump and aggregated other types (i.e., livestock, logging, etc.). In addition to truck-trailer classification, methods are developed to detect empty and loaded platform semi-trailers. K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Adaptive Boosting (AdaBoost), and Support Vector Machines (SVM) supervised machine learning algorithms are implemented on the field data collected on a freeway segment that includes over seven-thousand trucks. The results show that different trailer body types and empty and loaded platform semi-trailers can be classified with a very high level of accuracy ranging from 85% to 98% and 99%, respectively. To enhance the accuracy by which multiple LiDAR frames belonging to the same truck are merged, a new algorithm is developed to estimate the speed while the truck is within the field of view of the sensor. This algorithm is based on tracking tires and utilizes line detection concepts from image processing. The proposed algorithm improves the results and allows creating more accurate 2D and 3D truck profiles as documented in this thesis.

Copyright, 2020, by Olcay Sahin, All Rights Reserved.

This thesis is dedicated to my beloved wife, Demet,
for her tireless support and love.

ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this dissertation. I extend special thanks to my doctoral advisor Dr. Mecit Cetin for his guidance and support for both academic development and career.

I would also like to acknowledge the committee members who have guided me with their comments and suggestions: Dr. Sherif Ishak and Dr. Hong Yang.

Finally, I sincerely thank my family. Especially, my beloved wife Demet who supported me in this process while she was experiencing the challenges more than anybody else. I will always be thankful for her unending love and support. With a partner like you, I feel so fortunate. Our children Miray and Zeynep Eda have been around whenever I need a mental break. Thank you, my sweet ladies.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
 Chapter	
1. INTRODUCTION.....	1
1.1 Main contributions.....	5
1.2 Structure of the Dissertation.....	5
2. LITERATURE REVIEW.....	6
2.1 Vehicle Classification.....	6
2.2 Truck-trailer Classification.....	8
2.3 Detecting Empty Trips.....	9
3. DATA COLLECTION.....	10
3.1 Hardware Configuration.....	10
3.2 Software Configuration.....	12
3.3 Site Selection.....	15
3.4 Data Collection Schedule.....	17
3.5 Ground Truth Labeling.....	18
3.6 Classification Scheme Development.....	21
3.7 Summary.....	24
4. METHODOLOGY.....	25
4.1 Data Preprocessing.....	26
4.2 Constructing 2D or 3D Vehicle Profiles.....	31
4.3 Classification Algorithms.....	53
5. TRUCK-TRAILER CLASSIFICATION.....	55
5.1 Feature Extraction.....	55
5.2 Classification.....	61
5.3 Empirical Analyses and Results.....	65
5.4 Summary.....	70
6. DETECTING EMPTY AND LOADED PLATFORM SEMI-TRAILERS.....	71
6.1 Feature Extraction.....	71
6.2 Classification.....	73
6.3 Empirical Analyses and Results.....	74
6.4 Summary.....	76
7. CONCLUSION AND FUTURE WORK.....	78
REFERENCES.....	80
APPENDIX A.....	83

APPENDIX B	84
VITA	93

LIST OF TABLES

Table	Page
1. Common technologies for vehicle classification	2
2. Calendar for the collected LiDAR data in 2017	18
3. VIUS and Hernandez, Tok [36] Model Body Classification for Semi-Trailers for Existing VIUS Classes	23
4. Data labelled based on the HPMS.....	24
5. Pseudocode for Hough Transform line detection	42
6. Intermodal containers sample size	49
7. Statistic of estimated speed and intermodal container length by three-speed estimation methods.....	51
8. Features extracted for each truck and descriptive statistics	63
9. Training and testing data	65
10. Results from the classification algorithms on the test data	66
11. Confusion matrix (for SVM method), rows represent the actual class and columns the predicted class	68
12. Misclassified and correctly classified truck-trailers from the SVM model disaggregated by speed.....	69
13. Pearson's chi-squared tests for checking whether the classification results are affected when samples are disaggregated by speed.....	69
14. Results from the classification algorithms	75
15. Confusion matrix for each ML model. Rows are actual, and columns are predicted values from the models for the testing data.	76

LIST OF FIGURES

Figure	Page
1. Whole data collection system	11
2. Built waterproof case for electronic equipment and carrier to hold LiDAR and surveillance cameras	12
3. Wireshark output feature for automatic file creation	13
4. Sample .pcap files created with the selected naming format	14
5. (a) Data collection site, (b) LiDAR and surveillance cameras, (c) LiDAR mounted on the gantry pole before HRBT	15
6. (a) Horizontal LiDAR configuration, (b) sample scan from horizontally	16
7. (a) Vertical LiDAR configuration, (b) sample scan from vertically	17
8. Flow chart for extracting vehicle cloud point data from raw LiDAR data	19
9. Interface developed to manually label vehicle classes	21
10. Main components of the procedure for classifying truck trailers and detecting empty and loaded platform trailer from LiDAR data	25
11. (a) A full scan or frame from the LiDAR sensor. (b) After redundant points removed	27
12. Labeling of each LiDAR frame (rows) and beam (columns) belonging to individual vehicles numbered consecutively (numbers in the table)	28
13. Screenshot of saved LiDAR raw data for a vehicle	29
14. Equations used to calculate the height of a given point on a vehicle to the roadway surface represented by a plane	29
15. Before (a) and after (b) coordinate transformation	30

Figure	Page
16. Detection zone of the LiDAR marked on the pavement, top view	32
17. A 2D view of truck generated from each frame as it progresses through FOV (colors represent reflectivity)	35
18. Estimated and averaged distributed speed across the frames	36
19. Estimated and smoothed interpolated speed across the frames	37
20. Used cloud points for the axle trajectory in the region of interest for FHWA Class 9 semi-truck hauling a dry-van trailer	39
21. Axles trajectory for an FHWA Class 9 semi-truck hauling a dry-van trailer	40
22. First 100 lines based on the HT voting method.....	43
23. K-means identified dominant lines in the clusters.....	44
24. Estimated, k-means, and smoothed speed across the frame numbers	45
25. Speed values for reconstructing the vehicles profiles	46
26. Constructed 2D truck profiles using proposed three-speed estimation methods.....	47
27. Constructed 3D truck profiles using proposed three-speed estimation methods.....	48
28. Averaged constant speed by frame number for 40ft intermodal containers	50
29. 40ft Intermodal container count by the number of frames into three bins.....	50
30. Speed estimation methods tested on 40 ft intermodal containers. The number of LiDAR frames is less than 30	52
31. Speed estimation methods tested on 40 ft intermodal containers. The number of LiDAR frames greater and equal to 30.....	53
32. Extracted features annotated on a 3D profile of a refrigerated dry van.....	55
33. Trailer length and height calculation steps	58

Figure	Page
34. Residuals of standard deviation by side and top region	64
35. 36 Voxels from different views with a 3D truck cloud points. (a) Full View (b) Rear View (c) Front View	72

CHAPTER 1

INTRODUCTION

Vehicle classification plays a significant role in almost all aspects of transportation engineering and planning applications. In the mid-1980s, the Federal Highway Administration (FHWA) developed a standardized 13-category vehicle classification ruleset, which meets the needs of many traffic data user applications, such as highway and pavement design, performance monitoring, tolling, transportation planning, and freight planning and modeling, etc. The rule set is designed to classify visual descriptions of vehicles using axle-related metrics, such as the number of axles, axle spacing, number of trailers, and vehicle length with the available intrusive and non-intrusive equipment. The intrusive equipment includes inductive loops, road tubes, and piezo sensors, etc. The non-intrusive equipment consists of a video detection system, passive infrared, radar, etc. The complete list of standard and current data collection technologies can be seen in Table 1 [1]. In general, ruleset separates vehicles into categories depending on passenger vehicles or commodities. Many engineering and planning organizations use the generalized four bins category, which are cars, small trucks, large trucks, and multi-trailer trucks.

Table 1. Common technologies for vehicle classification

Axle-Based	Length-Based
Infrared (passive) (NI)	Dual inductive loops (I)
Laser radar (NI)	Inductive loops (loop signature) (I)
Piezo-electric (I)	Magnetic (magnetometer) (I)
Quartz sensor (I)	Video detection system (NI)
Fiber optic (I)	Microwave radar (NI)
Inductive Loop Signatures (I)	CW Doppler sensors (NI)
Capacitance mats (I)	
Bending plates (I)	
Load cells (I)	
Contact switch closures (e.g., road tubes)	
Specialized inductive loop systems	

Key: Non-Intrusive (NI), Intrusive (I), Source: (FHWA 2016)

However, the four-bins category is not sufficient for the estimation of pavement loads and freight planning and modeling. In 2003, the Transportation Research Board (TRB) Expert Task Group (ETG) on Long-Term Pavement Performance (LTPP) Traffic Data Collection and Analysis developed a new set of rules for classifying vehicles based on sensor outputs available from WIM systems. The rule set provides appropriate data for pavement loads analysis.

Researchers utilized data from such sensors to extract additional information such as travel times between two sensor locations and truck-flow patterns by anonymously reidentifying trucks [2, 3]. Being able to distinguish between different types of trucks is particularly important for freight planning and modeling. For example, long-haul and short-haul freight is typically transported on different truck types – tractor-trailers for the former and smaller rigid trucks for the latter. Furthermore, truck body types or trailer types may help reveal information about the commodity being carried. For example, refrigerated trailers are commonly used to transport perishable produce and meat products, tank trailers are for fuel and other liquid products, and specialized trailers carry livestock. It is obvious that not all commodity types can be easily

inferred from the externally observable characteristics of the truck or trailer; however, it is possible to narrow the possible types of commodities if the trailers are classified into distinct categories (e.g., car-transporter, tank, enclosed van, intermodal container, empty platform or trailer).

It should be mentioned that freight planning is a multifaced process that requires a variety of data [4]. Knowing truck body types along major truck corridors helps support different freight planning and analysis applications. To further motivate the potential value of classifying trucks into subcategories based on their trailers, two specific examples are given below.

- Impacts of capital investments and different policies (e.g., tolling, rush-hour restrictions for trucks) along a major truck corridor can be assessed in more detail if the truck body types are known. This will allow estimating which freight industry might be affected by such decisions. For example, if the corridor is serving a large (or negligible) number of intermodal containers, the impacts on containerized cargo will be significant (or negligible). This information will be helpful in engaging different stakeholders (e.g., ports, distribution centers) in the freight planning process. In this research, we show that intermodal containers can be detected with a relatively high accuracy within the FHWA Class 9 truck population.
- Knowing truck body types will help support the validation of commodity-based freight models. Commodity-based freight models convert commodity flows (in tonnage) between origins-and-destinations by assuming average payloads for trucks to produce an estimate of truck trips on the network. These estimated truck trips can be evaluated for accuracy at a more granular level if the truck body types are known. For example, gasoline and fuel oils are carried in tanks and logs in specialized trailers. Knowing the

number of trucks of each type at a network link will allow assessing how accurately these specific commodities are being modeled. This will ultimately improve the fidelity of freight modeling at a network level.

To adequately allocate resources for accommodating commercial freight delivery and distribution across the highway network, public agencies need to predict the quantity and type of commodities. The primary source for freight planning analysis in the USA is the Commodity Flow Survey (CFS) conducted by the Census Bureau every five years. In the commodity-based models, cargo weight is the unit of demand, and the information taken from CFS (e.g., weight, value) is distributed over suitable body types using specific payloads for loaded trucks. In addition to loaded trips, empty trips must then be determined and added to the flows of loaded trips to perform traffic assignment. Most of the state DOTs use a flat percentage of trips or weight-in-motion (WIM) counts to estimate empty trips for the freight modelling activities.

Multi-array LiDAR sensors are becoming widely available in the transportation field due to recent research and developments in automated driving technologies [5]. The raw point-cloud data from these sensors include distance to the target and intensity of the reflected light. From these data, it is possible to construct a 3D representation of the objects within the range of the sensor.

In this research, data from a Velodyne VLP-16 LiDAR are used to first classify truck trailers into ten types: (1) Enclosed or dry vans, (2) refrigerated vans, (3) 20 ft and (4) 40 ft intermodal containers, (5) 40 ft refrigerated container, (6) platforms, (7) tanks, (8) car transporter, (9) open top van/dump and (10) other. The category “other” includes livestock, logging and customized truck trailers, etc. Second, platform semi-trailers are classified as empty or loaded.

1.1 Main contributions

The main contributions of this research include the following: (i) demonstrating, for the first time, how multi-array LiDAR data could be utilized in classifying semi-trailers into ten subcategories; (ii) detecting empty and loaded platform semi-trailers; (iii) developing effective methods for extracting useful features from 3D LiDAR data; and (iv) developing classification algorithms that yield relatively high accuracies.

1.2 Structure of the Dissertation

Chapter 2 provides a review of the literature. In Chapter 3, we will describe data collection, and in Chapter 4, we provide details of methodological steps employed in this research. In Chapter 5 and Chapter 6, the extracted features for the classification of the ten subcategories and empty and loaded platform semi-trailers classification will be explained, respectively. Their results are presented at the end of each chapter. Chapter 7 includes conclusions and suggestions for future work.

CHAPTER 2

LITERATURE REVIEW

The literature review is organized into three main categories: 1) vehicle classification, 2) truck-trailer classification, 3) detecting empty trips. In the vehicle and truck-trailer classification sections, the studies using traditional sensors, image processing, and LiDAR data are summarized. To the best of our knowledge, there is no prior research on detecting empty trips using LiDAR data. However, there are some attempts to detect empty trips using WIM data and survey methods.

2.1 Vehicle Classification

As mentioned earlier, FHWA developed a rule set based on the vehicle's visual descriptions such as the number of axles, axle spacing, number of trailers, and vehicle length using the available sensors. These sensors are listed in Table 1.

The traditional sensors, namely, loop detectors, WIM sensors, and radars are widely used by state agencies and municipalities. The developed software (e.g., HI-COMM 100, IRD, PEEK VIPER and TOPS, etc.) [1] for classification using these sensors provides necessary information for most applications. However, researchers have recently investigated how these sensors could be used for monitoring traffic flow and estimating vehicle trajectories [6-8] and for evaluating highway safety issues [9-12]. Researchers are also exploring how GPS and smartphone data can be utilized for various transportation applications such as trajectory creation using smartphone inertial sensors[13, 14], vehicle speed and stop estimation using smartphone accelerometer data [15-17], modeling the impact of latent driving patterns on traffic safety using smartphone GPS data [18, 19], and feasibility of estimating commodity flows on highways with existing and emerging technologies [20].

In addition to traditional traffic sensors, camera sensors are also installed for instantly monitoring traffic flow. Researchers have used images from surveillance cameras [21, 22] to classify vehicles. The pictures from camera sensors are processed using several image processing algorithms (e.g., contrast enhancement, noise reduction, edge sharpening, edge detection, segmentation, etc.). One of the main challenges of vehicle classification is extracting the vehicle region from the background. Researchers use various methods to overcome this challenge. For example, Gupte, Masoud [23] uses optical flow, to separate background from foreground. Chung-Lin and Wen-Chieh [25] explored background subtraction and defined at intersections and on roadsides, low-altitude airborne platforms are used for vehicle classification [26, 27]. Coifman, McCord [26] describe the methods to determine level of service, average annual daily traffic, intersection operations, origin–destination flows on a small network, and parking lot utilization from data collected by an unmanned aerial vehicle. Khan and Cheng [27] used aerial images to match vehicles’ 3D models using Histogram of Oriented Gradients (HoG) [28] for classification of various vehicle types (e.g., sedan, SUV, truck, etc.). Using the camera sensors for tracking and classification is a challenging task due to the following reasons: occlusions, shadows, camera noise, changes in lighting and weather conditions. Researchers explored other sensors which are not sensitive to lighting and may be less sensitive to other environmental conditions (e.g., radar, LiDAR).

There is some limited research on classifying vehicles based on LiDAR data [29-34]. The majority of these works classify vehicles based on three major groups: (i) passenger vehicles, (ii) light duty trucks, and (iii) heavy duty trucks. Xiao et al. used data from a LiDAR onboard a vehicle that surveys parking spots along a street to detect and identify parked vehicles to support automated parking management applications [35]. In a related work, Lee and Coifman [30]

proposed the use of a multi-array LiDAR that is used as part of a portable, nonintrusive system for assessing the performance of vehicle classification stations [34]. Lee and Coifman [30] installed two multi-array LiDAR on top of a minivan with a distance of 4.6 ft [30, 34]. Then they extract features (e.g., length, height, middle drop) as input to a decision tree algorithm to classify motorcycle, passenger vehicle, passenger vehicle pulling a trailer, single-unit truck, single-unit truck pulling a trailer, and multi-unit truck. The developed decision tree algorithm correctly classifies non-occluded vehicles with 99.5% accuracy.

2.2 Truck-trailer Classification

Each truck trailer has its own shape, characteristics, and dimensions. The base of the truck-trailer classes was developed by the United States Census Bureau Vehicle Inventory and Use Survey (VIUS). VIUS captured the vehicle population in three main categories: passenger vehicles, single-unit trucks, and semi-tractor trailer combination trucks. According to the 2002 VIUS, there are 11 major body types of trucks (platform, van, auto transport, dump, grain bodies, garbage, livestock, logging, dry tank, liquid tank, and other) in the USA. VIUS is the only available source for the truck-trailer classification.

There are attempts to classify truck body types into subgroups relevant to the commodities being transported using data from traditional sensors and inductive loop signatures [36]. Unfortunately, to the best of our knowledge this is the only available study for truck-trailer classification using traditional traffic sensors.

Vatani Nezafat, Sahin [37] generated 2D images of trucks from LiDAR scans and classified them using convolutional deep neural networks. However, in that study only four trailer types were considered [37] and 3D LiDAR data were not fully exploited.

Asborno and Burris [38] implemented two single-beam low-cost LiDARs to classify truck body types into five distinct five-axle tractor-trailer body types, namely: van and container, platform, low-profile trailer, tanks, and hopper and end dumps [38]. The LiDAR employed by these researchers has a single beam that scans the scene at a fixed height and angle. Therefore, it provides distance to the truck body only at the preset height. This information helps extract additional data about truck body types, but it is not possible to generate 3D complete profiles. In terms of classification accuracy, the authors report an average true positive rate of 81% [38]. More research is needed to fully utilize the rich multi-array LiDAR data for refined classification schemes, especially for different types of truck trailers.

2.3 Detecting Empty Trips

In the commodity-based models, cargo's weight is the unit of demand, and the information taken from CFS (e.g., weight, value) is distributed over suitable body types using specific payloads for loaded trucks. In order to better quantify empty trips, researchers developed empty trip models [39, 40] and used a supply chain logistical approach [41] or behavioral and operational characteristics of the carriers [42]. FDOT Transportation Data and Analytics office performed a study regarding truck empty backhaul [43]. In this study, WIM data is analyzed to estimate whether a truck is an empty backhaul. Nevertheless, there is no systematic way to identify empty trips so that such freight models can be calibrated and validated. However, in this research we investigate how LiDAR sensors could be utilized to distinguish between different trailer types and detect empty and loaded platform semi-trailers.

CHAPTER 3

DATA COLLECTION

The data needed for this thesis are collected at a major freeway in Hampton Roads, VA. Velodyne VLP-16 LiDAR and a set of high definition (HD) surveillance cameras are used for data collection. Considering the continuous data collection from two independent systems and their size, an efficient hardware and software configuration is needed. The hardware and software configuration, selected site, handling data from two different systems, and ground truth labeling processes are described in this section.

3.1 Hardware Configuration

The data needed to conduct this research come from a LiDAR unit and a set of surveillance cameras. There is also other hardware for a complete data collection system, e.g., power supply, a computer, ethernet switch, external hard drives, and internet-enabled smartphone for remote desktop.

In order to collect data from Velodyne LiDAR, an external CPU unit (e.g., computer) and software are necessary. The Velodyne VLP-16 LiDAR is programmed to store collected point cloud data in the packet (.pcap) format, using an ethernet networking interface. Using this interface, LiDAR can be attached to a local area network, and it can be accessed through a computer that is in the same network. The advantage of using a local area network is that other hardware can be also attached to the same network switch e.g., computer, surveillance cameras, and network hard drive. Thus, the whole system can be controlled using a single computer.

Since a local area network already is created, a surveillance system compatible with the network connection can be used. There are many choices available in the market. For this research, we used a Swann 4 camera 5MP Super HD surveillance system. The advantage of this

system is that the timestamp can be synchronized with the computer timestamp which is connected to the same network. The surveillance system automatically renames the recorded files with a current timestamp which will be used to match video to LiDAR files.

The designed data collection system will be on-site and may not be accessible all the time. Therefore, a remote connection is needed to check system health. For this purpose, a high-speed internet-enabled smartphone can be attached to a computer and a remote connection can be established through the smartphone to monitor if the system is running. We can visit the data collection site only when a problem occurs. The designed data collection system is illustrated in Figure 1.

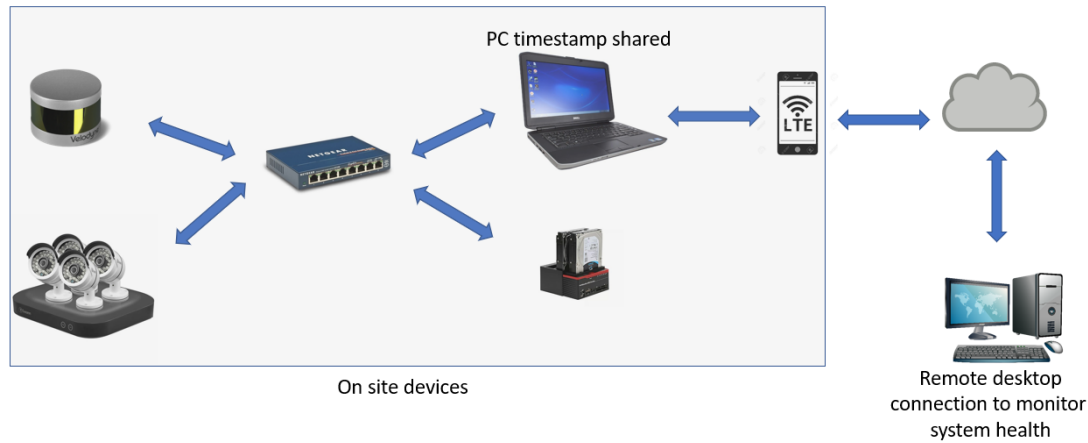


Figure 1. Whole data collection system

All sensitive equipment in the data collection system e.g., laptop, ethernet switch, hard drives, etc., must be located in a waterproof case to protect from rain, snow, and dust. LiDAR

and surveillance cameras are placed onto a carrier. The built waterproof case and carrier can be seen in Figure 2.



Figure 2. Built waterproof case for electronic equipment and carrier to hold LiDAR and surveillance cameras

3.2 Software Configuration

3.2.1 LiDAR Software

Velodyne VLP-16 LiDAR manufacturer developed the Veloview software to visualize and record the LiDAR data. Unfortunately, this software does not support automated data collection. A user must start and stop a recording manually. Also, if data collection runs on for many hours, data will be lost in circumstances such as power loss or exceeding file storage space. Therefore, other software must be considered for data collection.

Since LiDAR is accessible with a network interface, the Wireshark network protocol analyzer program can be used to record data. This software allows recording data from any network interface (e.g., ethernet, Wi-fi, Bluetooth). One of the advantages of using this software is that a new file is automatically created after a defined size (e.g., 1GB). In this study, this software will be used with an automatic file creation feature. The automated file creation threshold is set at 1GB as seen in Figure 3. Wireshark automatically adds the timestamp when the file is created. The created file naming format can be seen in Figure 4.

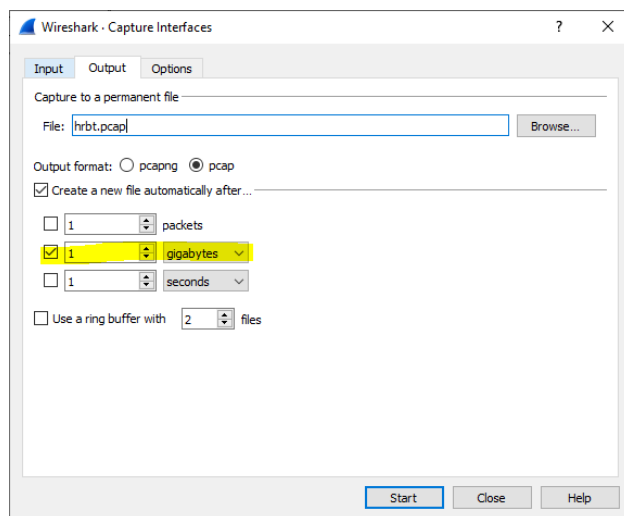


Figure 3. Wireshark output feature for automatic file creation




















Name	Date modified	Type	Size
 hrbt_00141_20171027074440.pcap	10/27/2017 8:00 AM	PCAP File	976,564 KB
 hrbt_00142_20171027080045.pcap	10/27/2017 8:16 AM	PCAP File	976,564 KB
 hrbt_00143_20171027081650.pcap	10/27/2017 8:32 AM	PCAP File	976,564 KB
 hrbt_00144_20171027083254.pcap	10/27/2017 8:48 AM	PCAP File	976,563 KB
 hrbt_00145_20171027084859.pcap	10/27/2017 9:05 AM	PCAP File	976,563 KB
 hrbt_00146_20171027090504.pcap	10/27/2017 9:21 AM	PCAP File	976,564 KB
 hrbt_00147_20171027092109.pcap	10/27/2017 9:37 AM	PCAP File	976,564 KB
 hrbt_00148_20171027093714.pcap	10/27/2017 9:53 AM	PCAP File	976,563 KB
 hrbt_00149_20171027095318.pcap	10/27/2017 10:09 AM	PCAP File	976,564 KB
 hrbt_00150_20171027100923.pcap	10/27/2017 10:25 AM	PCAP File	976,564 KB
 hrbt_00151_20171027102528.pcap	10/27/2017 10:38 AM	PCAP File	976,564 KB
 hrbt_00152_20171027103802.pcap	10/27/2017 10:54 AM	PCAP File	976,563 KB
 hrbt_00153_20171027105407.pcap	10/27/2017 11:10 AM	PCAP File	976,564 KB
 hrbt_00154_20171027111012.pcap	10/27/2017 11:26 AM	PCAP File	976,563 KB
 hrbt_00155_20171027112616.pcap	10/27/2017 11:42 AM	PCAP File	976,564 KB
 hrbt_00156_20171027114221.pcap	10/27/2017 11:58 AM	PCAP File	976,564 KB
 hrbt_00157_20171027115826.pcap	10/27/2017 12:14 PM	PCAP File	976,564 KB
 hrbt_00158_20171027121431.pcap	10/27/2017 12:30 PM	PCAP File	976,563 KB
 hrbt_00159_20171027123036.pcap	10/27/2017 12:46 PM	PCAP File	976,564 KB

Figure 4. Sample .pcap files created with the selected naming format

3.2.2 Video Software

The purchased surveillance system has its own operating system which records files in a DVR box. The system allows us to set up desired recording times, image quality, location, timestamp, etc. In this research, we are interested in only daytime recording because the video will be used for ground truth purposes, and at nighttime it is difficult to observe vehicle types. Therefore, recording times are set for the daytime and picture quality set to 4K at 10 hertz. The video files are saved onto the local DVR system, and computer time has been used through the network connection.

3.3 Site Selection

The equipment is installed on a gantry pole, carrying a variable message sign across I-64 westbound in Hampton Roads, VA (Figure 5-a). The road segment at this location has two lanes (Figure 5-c), and the site is about one mile upstream of the Hampton Roads Bridge-Tunnel (HRBT). Trucks are prohibited from traveling in the left lane. Therefore, the LiDAR sensor is configured to capture vehicles in the rightmost lane. As illustrated in Figure 5-b, the LiDAR sensor is 22 ft above ground and 20 ft away from the travel lane. The LiDAR sensor and the surveillance cameras are oriented to get good coverage of the vehicles traveling in the right lane.

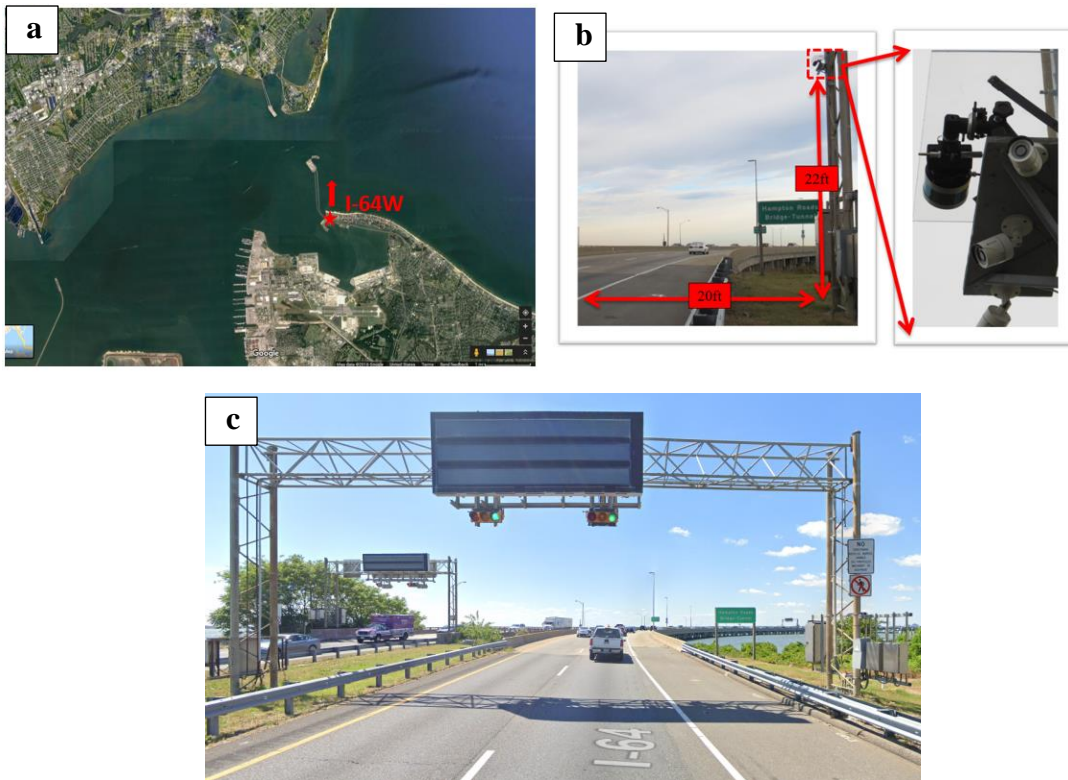


Figure 5. (a) Data collection site, (b) LiDAR and surveillance cameras, (c) LiDAR mounted on the gantry pole before HRBT

The LiDAR sensor can be installed in a vertical or horizontal (or any other angle) scanning mode depending on the application. If mounted horizontally (Figure 6-a) on the roadside, a sensor covers a maximum of 164 ft (Figure 6-b) from the sensor in the longitudinal direction of the roadway, but it will not result in a dense set of points for each vehicle observed. In the vertical configuration (Figure 7-a), a sensor covers about 16 ft of the longitudinal section of the rightmost lane and provides more dense points per vehicle. Most passenger vehicles can fit in this range, but vehicles longer than 16 ft (Figure 7-b) will not. Therefore, multiple scans or frames need to be combined to create the full 3D or 2D profiles of trucks. For this study, the LiDAR is configured in the vertical orientation.

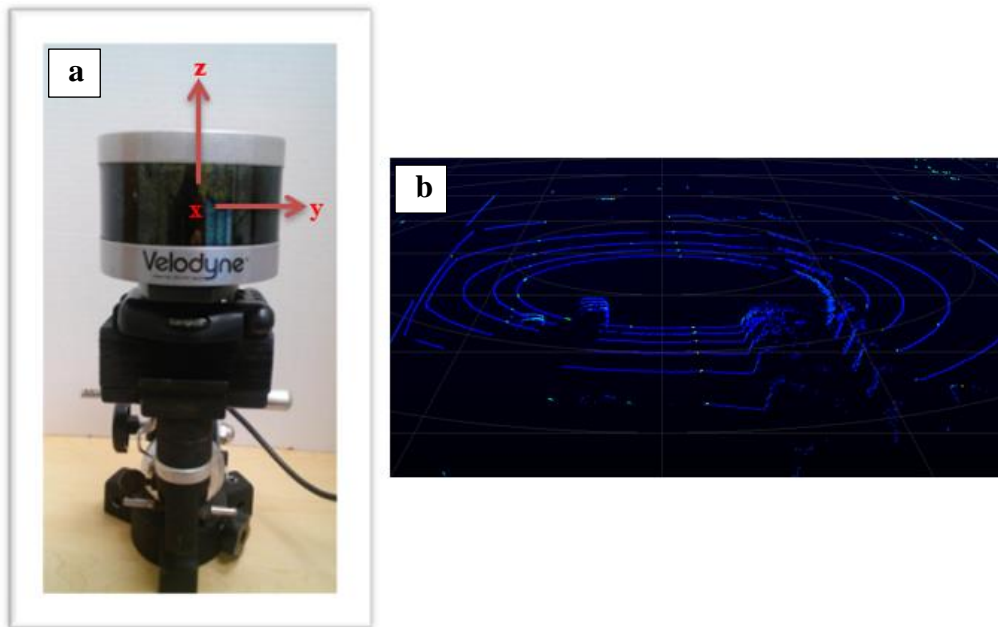


Figure 6. (a) Horizontal LiDAR configuration, (b) sample scan from horizontally

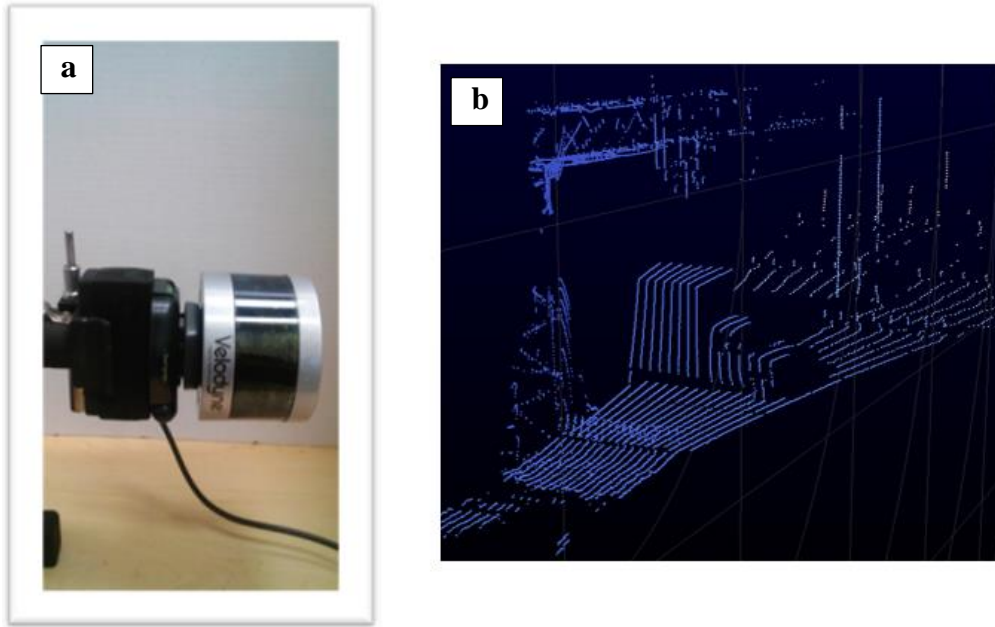


Figure 7. (a) Vertical LiDAR configuration, (b) sample scan from vertically

3.4 Data Collection Schedule

The LiDAR and camera systems were installed in the field on October 25, 2017, in the westbound direction on I-64 and set to record data during the day from 7:00 AM until 6:00 PM. The system started to collect data on October 27 and was in place until December 12, 2017. Table 2 shows the schedule of the collected data. Due to power outages and equipment malfunctions, data from several days within this period were not collected. This is indicated by empty cells in Table 2. Cells highlighted in green indicate the periods where both LiDAR and camera systems were operational whereas yellow cells indicate when only cameras were working. Rainy days/times are indicated by an “R.” LiDAR data from rainy days were not included and analyzed since LiDAR’s reading is affected by rain (e.g., spurious returns from plumes of rainwater). Overall, the collected data amount to 3.2 TB of .pcap LiDAR files and

18.5 TB of video files. LiDAR data were collected over 36 days, as indicated in the last row of the table.

Table 2. Calendar for the collected LiDAR data in 2017

		October							November																					December																					
Day#	AM	PM	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12		
			R	R	R											R	R	R	R																																
Legend:			LIDAR and Camera							Camera							No Recording							R:Rain																											

3.5 Ground Truth Labeling

The data ground truth procedure involved preprocessing the LiDAR and video data then identifying the vehicle configuration and body type from each vehicle through a customized user interface.

3.5.1 LiDAR Data

At the end of the data collection there were around 3,200 LiDAR files. Each file contains 15 minutes of a scan containing around 10,000 frames. Veloview software can be used to manually open and extract these LiDAR files, but it is time-consuming to perform this process manually. Therefore, a custom script was developed in the Python programming language which opens and extracts LiDAR files automatically using parallel computing. The flow chart of the

developed script can be seen in Figure 8. A method is developed to extract only a vehicle's cloud points from the LiDAR files. This method will be explained in the methodology section.

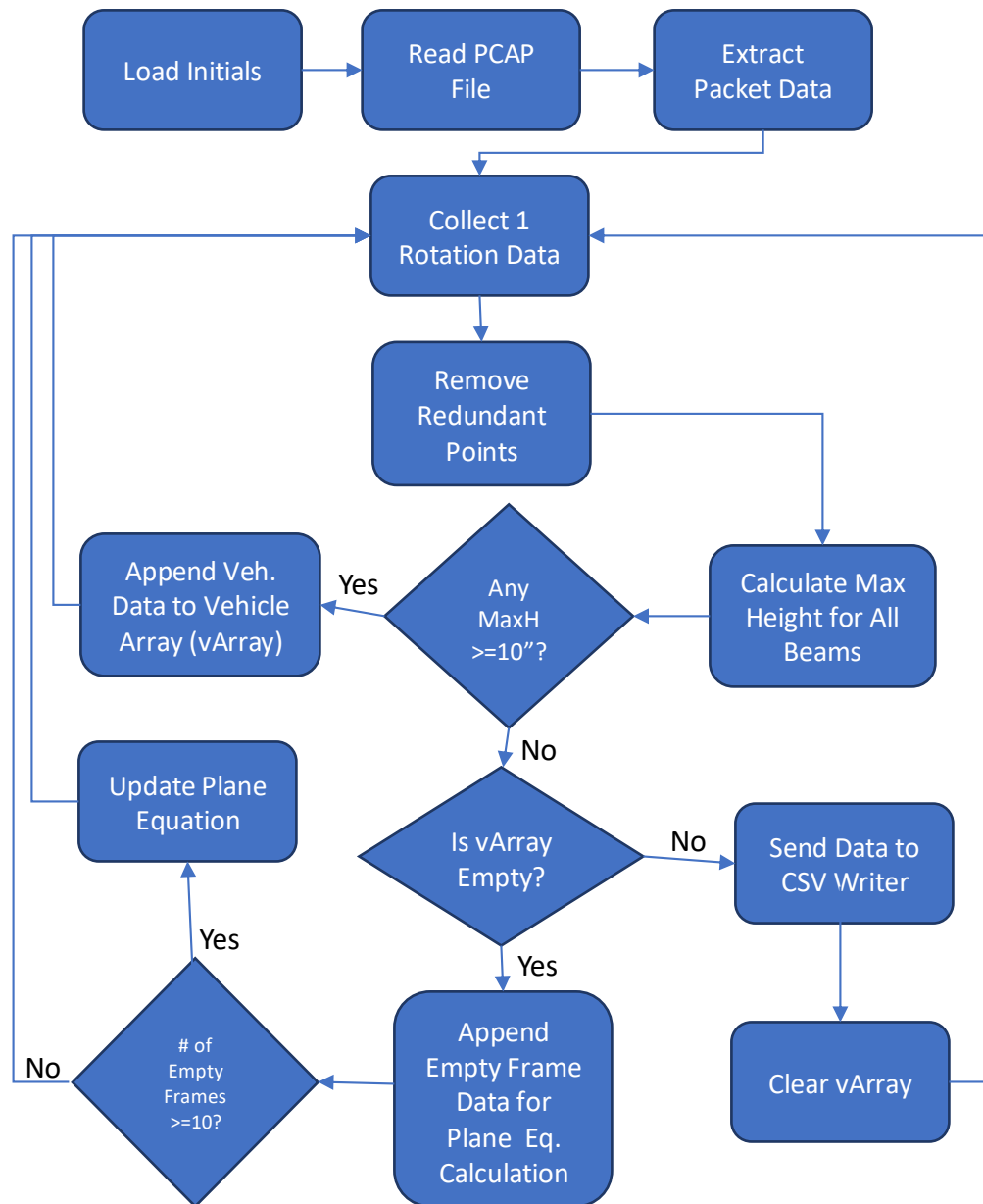


Figure 8. Flow chart for extracting vehicle cloud point data from raw LiDAR data

3.5.2 Video Data

Four surveillance cameras were used to collect video data. Three of these cameras were installed directly below the LiDAR sensor (see Figure 2). The other camera was installed about 10 ft from the ground. The recordings were programmed to start at 7:00 AM till 6:00 PM every day. As stated before, the surveillance cameras used computer time to facilitate data synchronization and identify the same vehicle within the LiDAR data. The frequency of video data was set at 10Hz which is the same as the LiDAR settings.

Since LiDAR and video systems are set at the same frequency, every single vehicle's picture can be extracted using the information gathered from LiDAR data. The picture extraction can be done using the following process. To begin with, the first vehicle was selected from the LiDAR file for a day. Then, this vehicle was found in the video file. Second, the timestamp for this specific vehicle from the video file was recorded. The rest of the vehicle's video timestamp can be found since headway information is known for each vehicle from the LiDAR data. Thus, the picture extraction process is automated using the FFMPEG [44] tool.

3.5.3 Customized User Interface

To ease the ground truth process for each vehicle's video and LiDAR information, a unique identification number is stored in a PostgreSQL relational database. A data lake is created for storing all related files such as raw and extracted LiDAR files, video, and picture files.

A software user interface shown in Figure 9 is developed with RShiny framework. Using the developed UI, each truck-trailer configuration is manually entered based on Hernandez and Tok's [36] study that defined truck-trailer body classifications. The user interface was linked to the relational database to update vehicle configuration for the selected vehicle.

The user interface contains 3 columns. The leftmost column contains date selection, length filtration based on the Highway Performance Monitoring System (HPMS), and a selectable record table in the bottom left corner. The middle column contains vehicle images extracted from video and constructed 2D LiDAR points. The rightmost column contains basic information of the vehicle (e.g., vehicle id, date and time, number of LiDAR frames, max-height, and max length) and selectable vehicle body configurations that will be explained in the next section.

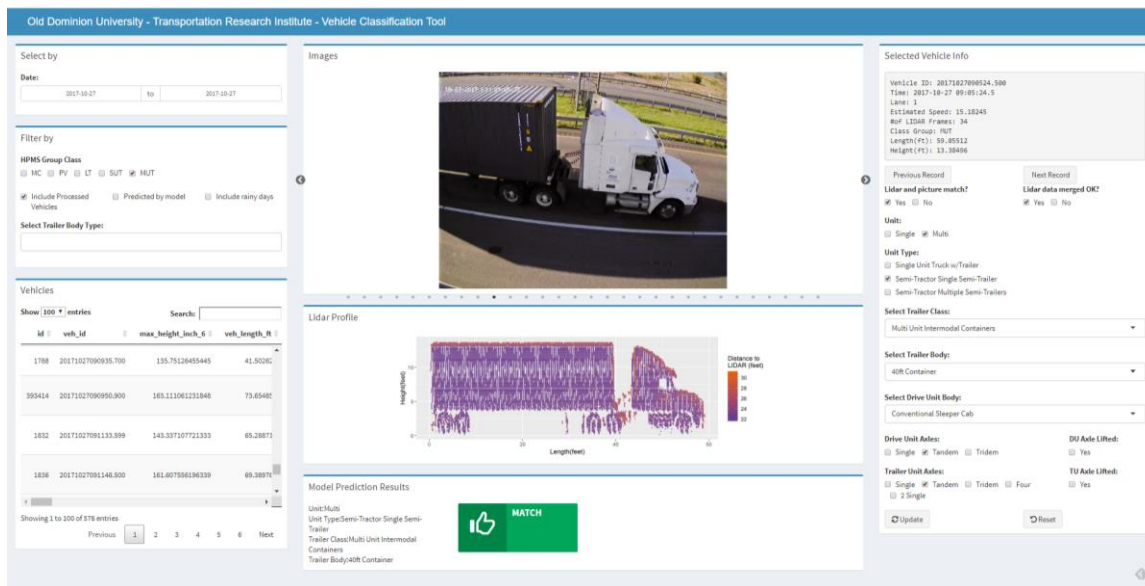


Figure 9. Interface developed to manually label vehicle classes

3.6 Classification Scheme Development

The ultimate goal of this study is classifying truck trailers into sub-categories (e.g., dry van, container, tank, platform, and specialty such as dump, automobile carrier, etc.). The base of the truck-trailer classes was developed by the United States Census Bureau Vehicle Inventory

and Use Survey (VIUS). VIUS captured vehicle population in three main categories: passenger vehicles, single-unit trucks, and semi-tractor trailer combination trucks. As of writing this dissertation, the most recent available survey was performed in 2002. The Census Bureau announced that a new vehicle survey will be conducted in 2022. In addition to VIUS, Hernandez and Tok [36] developed a model for truck-trailer body classification using WIM and inductive loop signatures. They introduced additional truck-trailer body classification on top of the VIUS. The comparison of truck-trailer body classification between VIUS and Hernandez and Tok's [36] study can be seen in Table 3. In this research, the categories proposed by Hernandez and Tok [36] for truck-body classification will be used.

Table 3. VIUS and Hernandez, Tok [36] Model Body Classification for Semi-Trailers for Existing VIUS Classes

Category	VIUS Body Class	Hernandez, Tok [36] Body Class
Van	Van, basic enclosed (dry cargo)	Enclosed van
	Van, insulated non-refrigerated	Skirted enclosed van
	Van, drop frame (excluding livestock)	Drop frame van
	Van, insulated refrigerated	Reefer enclosed van
Tank	Tank, dry bulk Tank, liquids or gases	Hot product tank
		Deep drop tank
		Food grade tank
		Petroleum tank
		Chemical tank
		Crude oil tank
		Air compression tank
		Propane tank
		Pneumatic Tank
Platform	Flatbed, platform, etc.	Basic platform
		Platform with devices
Specialty	Dump (including belly or bottom dump)	Low boy platform
		Bottom/Belly dump
		Bulk waste transport
	Livestock (including livestock drop frame)	End dump
		Livestock
		Curtainside van
	Mobile home toter	*
	Open tops (vans, low side grain, fruit, etc.)	Open top van
	Pole, logging, pulpwood, or pipe	Pole, logging, pulpwood, or pipe
	Automobile Carrier	Automobile transport
	Beverage	Beverage
	Trailer mounted equipment	*
	**	Hopper
	**	Agricultural van
Intermodal Containers	**	Container chassis
	**	40 ft container
	**	40 ft refrigerated container
	**	20 ft container
	**	20 ft cont. on 40 ft chassis

Table 3. Continued

Category	VIUS Body Class	Hernandez, Tok [36] Body Class
	**	53 ft container
Small Trailers	**	Recreational vehicle trailer
	**	Towed vehicle
	**	Small trailer/dolly
	* Not included in the model body classification scheme ** Not included in VIUS	

Note. Reprinted from “*Integration of Weigh-in-Motion and Inductive Signature Data for Truck Body Classification*”, by Hernandez, S., 2014, *Dissertation*, p. 67

3.7 Summary

Hardware and software systems are developed to collect data from a 2-lane section of I-64W. For each vehicle travelling in the rightmost lane, its information is recorded and linked to the developed user interface for manual labeling. Around 365,000 vehicles passed in the rightmost lane through the LiDAR field of view (FOV) area between 7:00 AM and 5:00 PM. The HPMS vehicle classification scheme applied to collected vehicle data for distinguishing the vehicle classes. Multi-unit-truck (MUT) configurations are manually labeled using Hernandez and Tok's [36] study on truck-trailer body classification scheme. The number of vehicles by HMPS classification can be seen in Table 4. The labeled MUT vehicles' trailer body types are also in Appendix A.

Table 4 Data labelled based on the HPMS

HPMS Class	Samples Collected
Motorcycle (MC)	349
Passenger Vehicle (PV)	292,832
Light Truck (LT)	38,406
Single Unit Truck (SUT)	18,570
Multi-Unit Truck (MUT)	15,182

CHAPTER 4

METHODOLOGY

LiDAR technology has been primarily used in geology, forestry, and mapping applications, and recently it has been deployed in automated vehicles to recognize objects surrounding the vehicle. LiDAR offers additional sensing capabilities beyond what is available with more traditional sensors. From the cloud points, 3D profiles of trucks can be generated, and these profiles allow extracting useful features for classification. In general, LiDAR can provide various measurements of interests such as trailer height and length to support applications beyond vehicle classification. For example, beyond truck classification, the LiDAR sensors can measure truck heights in real time to support monitoring over-height trucks at locations with height restrictions (e.g., underpasses and tunnels). As mentioned earlier, we use data from a LiDAR sensor mounted in a side-fire configuration on a roadside pole. Figure 10 shows the major steps in the overall procedure. Data preprocessing, vehicle profile construction, and classification procedures will be explained in this chapter. Feature extraction for truck-trailer classification and detecting empty and loaded platform trailers will be explained in Chapter 4 and Chapter 5, respectively.



Figure 10. Main components of the procedure for classifying truck trailers and detecting empty and loaded platform trailer from LiDAR data

4.1 Data Preprocessing

For data collection, we used the VLP-16 LiDAR unit manufactured by Velodyne [45]. This sensor comes with 16 beams, which covers a 30° view angle with 360° rotation around its internal z-axis. The LiDAR frequency is set at 10 Hz, which provides a very rich cloud point dataset. The unit can be installed in a vertical or horizontal (or any other angle) scanning mode depending on the application. If it is mounted horizontally (Figure 6) on the roadside, it covers a maximum of 164 ft from the sensor in the longitudinal direction of the roadway, but it will not result in a dense set of points for each vehicle observed. In the vertical configuration (Figure 7) for the selected site, it covers a 16 ft longitudinal section of the rightmost lane and provides more dense points per vehicle. Most passenger vehicles can fit in this range, but vehicles longer than 16 ft will not. Therefore, multiple scans or frames need to be combined to create the full 3D or 2D profiles of trucks. For this research, the LiDAR is configured in the vertical orientation. To create a profile of the vehicles from the collected data, redundant data points should be removed and LiDAR frames should be identified for the same vehicle. These will be explained in the next subsections.

4.1.1 Remove Redundant Data Points

Since trucks are traveling only in the right lane, LiDAR points reflected from objects elsewhere can be excluded from the dataset. Thresholds were established to eliminate these redundant data points. Figure 11-a shows a complete LiDAR scan, whereas Figure 11-b has the remaining data points after removing the redundant data. All the analyses are performed with the subset of points belonging to vehicles traveling in the right lane, as in Figure 11-b.



Figure 11. (a) A full scan or frame from the LiDAR sensor. (b) After redundant points removed.

4.1.2 Identify LiDAR Frames for the Same Vehicle

As mentioned earlier, there is more than one frame per vehicle. Before constructing a 2D or 3D profile of a vehicle from raw LiDAR data, all the frames which belong to a vehicle must be labeled. It can be accomplished as consecutively numbering all vehicles passing under the LiDAR detection zone and assigning each LiDAR beam and frame to the corresponding vehicle.

The existence of a vehicle in the LiDAR view area is identified in the following manner. First, all redundant data points are removed. Second, a plane equation (see Figure 14) is found by fitting a surface plane to the collected data when there is no vehicle in the LiDAR viewing area. Third, the height of each point is computed with respect to the roadway surface using the previously found plane equation. A method for the height calculation will be discussed in the next sub-section. Then all the data points for height are computed. Since points height is known, a threshold can be applied to detect a vehicle when entering and departing the LiDAR view area. Then, all corresponding frames and LiDAR beams are assigned a unique identifier for the vehicle. Figure 12 shows a sample table demonstrating how each LiDAR frame (rows) and beam (columns) belonging to individual vehicles are numbered (numbers in the table).

	0	2	4	6	8	10	12	14	1	3	5	7	9	11	13	15
6994	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6995	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6996	169	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6997	169	169	169	169	0	0	0	0	0	0	0	0	0	0	0	0
6998	169	169	169	169	169	169	0	0	0	0	0	0	0	0	0	0
6999	169	169	169	169	169	169	169	169	169	0	0	0	0	0	0	0
7000	169	169	169	169	169	169	169	169	169	169	169	169	0	0	0	0
7001	169	169	169	169	169	169	169	169	169	169	169	169	169	169	169	0
7002	0	169	169	169	169	169	169	169	169	169	169	169	169	169	169	169
7003	0	0	0	0	169	169	169	169	169	169	169	169	169	169	169	169
7004	0	0	0	0	0	0	0	169	169	169	169	169	169	169	169	169
7005	0	0	0	0	0	0	0	0	0	0	169	169	169	169	169	169
7006	170	170	0	0	0	0	0	0	0	0	0	0	0	169	169	169
7007	170	170	170	170	0	0	0	0	0	0	0	0	0	0	0	169
7008	170	170	170	170	170	170	170	0	0	0	0	0	0	0	0	0
7009	170	170	170	170	170	170	170	170	170	0	0	0	0	0	0	0
7010	170	170	170	170	170	170	170	170	170	170	170	170	0	0	0	0
7011	0	0	170	170	170	170	170	170	170	170	170	170	170	170	170	0
7012	0	0	0	0	0	170	170	170	170	170	170	170	170	170	170	170
7013	0	0	0	0	0	0	0	170	170	170	170	170	170	170	170	170
7014	0	0	0	0	0	0	0	0	0	0	170	170	170	170	170	170
7015	0	0	0	0	0	0	0	0	0	0	0	0	170	170	170	170
7016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	170
7017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12. Labeling of each LiDAR frame (rows) and beam (columns) belonging to individual vehicles numbered consecutively (numbers in the table)

After identifying the frames and beams for every vehicle, the raw data was saved into a single CSV file. This file contains x-y-z cartesian coordinates, timestamp, frame numbers, and other information about the specific data point. A sample file that belongs to a vehicle can be seen in Figure 13. The name of the CSV file is assigned as “*vehid_316* *hrbt_00003_20171207093503 (Frames 1299-1311).csv*” which contains information about the source LiDAR file, frame numbers, and vehicle order in the LiDAR file. The vehicle specific information is also recorded in a log file for further processing.

	A	B	C	D	E	F	G	H	I	J	K
1	x	y	z	intensity	laser_id	azimuth	dist_meter	timestamp	height_inch	frame_number	
2	4.195779	9.723798	-2.83769	1	0	2334	10.964	289442567	72.09166144	1299	
3	4.186353	9.614185	-2.80974	1	0	2353	10.856	289442622.3	72.55758754	1299	
4	4.219888	9.599513	-2.80974	1	0	2373	10.856	289442677.6	71.25668855	1299	
5	4.272177	9.627103	-2.82216	1	0	2393	10.904	289442732.9	69.18375542	1299	
6	4.327975	9.657216	-2.83562	1	0	2414	10.956	289442788.2	66.97117641	1299	
7	4.342221	9.519326	-2.80353	1	0	2452	10.832	289442898.8	66.53499502	1299	
8	4.364113	9.479544	-2.79628	1	0	2472	10.804	289442954.1	65.71261057	1299	
9	4.327173	9.313581	-2.75176	1	0	2492	10.632	289443009.4	67.30654831	1299	
10	4.329313	9.233702	-2.73261	12	0	2512	10.558	289443064.7	67.29346673	1299	

Figure 13. Screenshot of saved LiDAR raw data for a vehicle

4.1.2.1 Height Calculation

The height of a LiDAR point from the roadway surface is needed to create vehicle profiles. The height calculation can be accomplished by using basic geometry and the equation of a point to a plane (i.e., roadway surface) – see Equation (1). The equation can be found by fitting a surface plane to the LiDAR data which returns from the roadway surface when there is no vehicle.

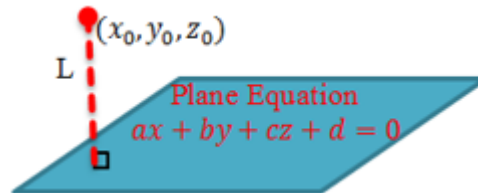


Figure 14. Equations used to calculate the height of a given point on a vehicle to the roadway surface represented by a plane

$$H = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (1)$$

4.1.3 Coordinate Transformation

The LiDAR sensor provides the position of each point in its 3D coordinate system. This coordinate system may not be fully aligned with the travel lane. Therefore, the raw data are transformed into a new coordinate system where the x-axis is along the travel direction, the y-axis is perpendicular to the roadway surface, and the z-axis is in the lateral direction. This transformation is simply accomplished by identifying unit vectors along these three directions. Then, the points are simply transformed into the new coordinate system by employing a rotation matrix, a commonly used coordinate transformation method.

As seen in Figure 15-a, a truck enters the field of view of the LiDAR. After the coordinate transformation is applied to the raw data, it is recreated as shown in Figure 15-b with all the original data preserved. Working in the new coordinate system facilitates the remaining steps where speed is estimated for merging multiple frames as explained next.

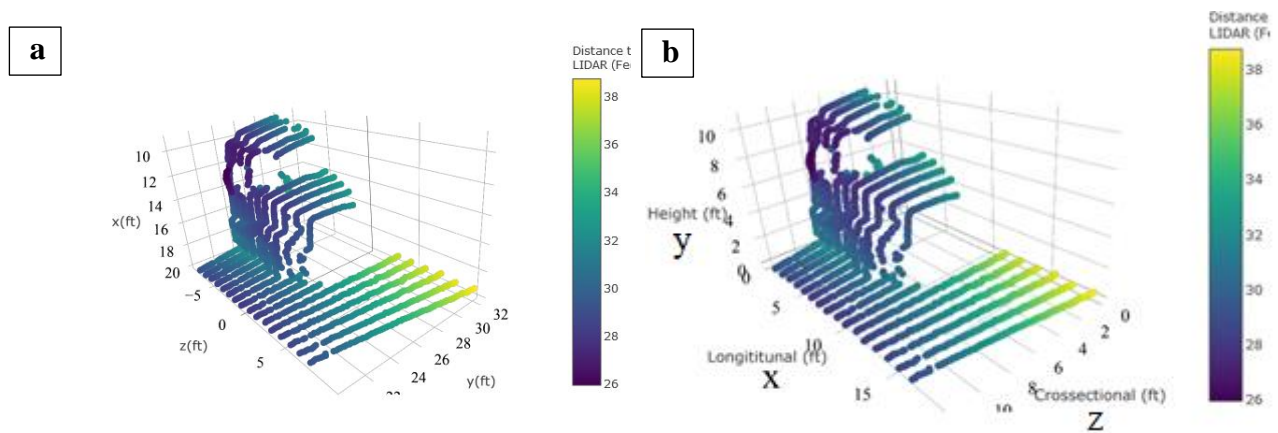


Figure 15. Before (a) and after (b) coordinate transformation

4.2 Constructing 2D or 3D Vehicle Profiles

As explained before, the entire truck does not fit within the detection zone or the LiDAR FOV. Therefore, to generate the full truck profile, multiple frames need to be merged. It can be done if the speed at which the truck is traveling is known. Unfortunately, at the data collection site, there was no speed measurement instrument available. However, with the LiDAR data, speed can be estimated since we know the length of the field of view and the LiDAR scan rate. Three different speed estimation methods will be explained in the next subsection.

In this section, first the LiDAR FOV will be explained in detail. Afterward, the speed estimation methods will be discussed, and the results for each method will be presented at the end.

4.2.1 LiDAR FOV

As indicated before, the LiDAR sensor scans ten times in one second with its 16 beams to measure depth. The labeled beams can be seen in Figure 16. The numbering of the beams shown is how it is defined by the manufacturer. Since laser beams emanate from the LiDAR sensor at 2° angles, the spacing between adjacent beams increases with increasing distance from the sensor. Figure 16 shows how the LiDAR's detection zone looks on the pavement and its dimensions when there is no vehicle. Along the longitudinal direction, the detection zone is 15' and 20', corresponding to 12 inches and 16 inches of spacing between consecutive beams, at the near and far sides of the lane, respectively. A truck traveling around the speed limit of 55 mph (approximately 80 fps) will be in the detection zone for about 1 sec if the truck length is 60 ft or 1.2 sec if the truck is about 80 ft long. This means that there will be 10-12 LiDAR frames, where at least some part of the truck is within the detection zone (see Figure 17). However, a specific

point on the truck (such as bumper cover) will cross the detection zone in about 0.2 sec. It means that there will be two frames where a specific point of the truck is within the detection zone. One can estimate vehicle speed using these frames. Speed estimation methods will be explained in more detail in the next section.

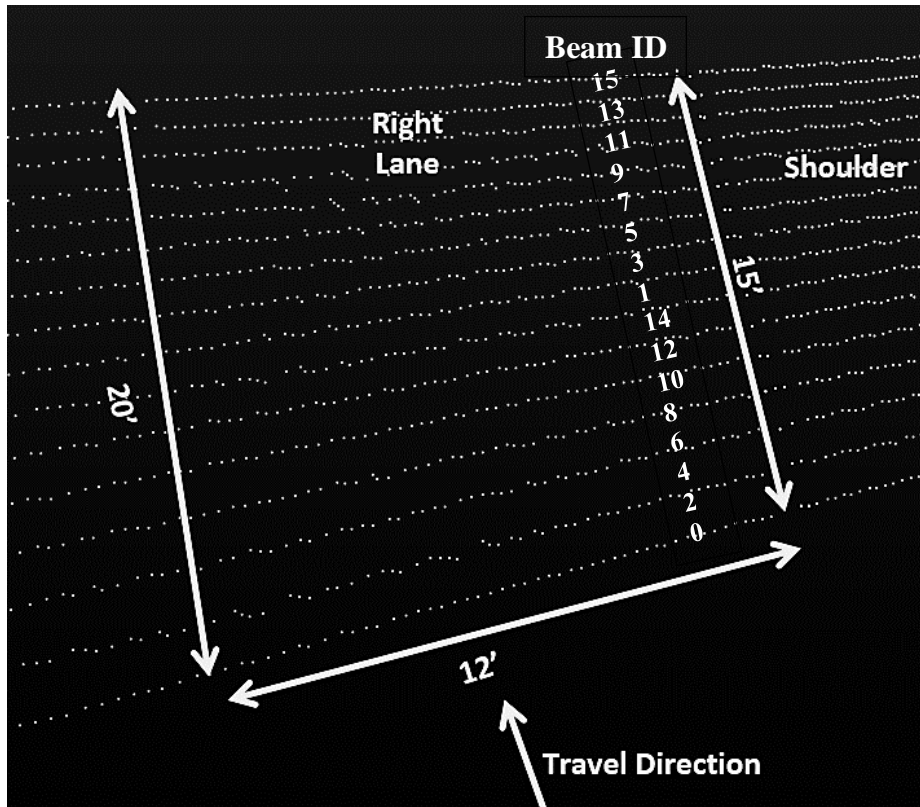


Figure 16. Detection zone of the LiDAR marked on the pavement, top view

4.2.2 Speed Estimation Methods

Based on freeway traffic, one can consider the speeds less than 20 mph as stop-and-go traffic, between 20 mph and 40 mph as congested, and greater than 40 mph as free-flow.

Constructing the vehicles' 2D or 3D profiles using a constant speed value may not work for

different traffic conditions. For example, if the vehicle speed is varying in LiDAR FOV, using a constant speed can distort the construction of the vehicle's profile. In this research, three different speed estimation methods are tested to construct vehicle profiles using LiDAR cloud points in various traffic conditions. These methods are:

1. Using a Constant Speed value (CS)
2. Speed estimation by Interpolation (IS)
3. Speed estimation by tracking Axles (AS)

For the first two methods (CS and IS), the speed can be estimated using the first two or more consecutive frames and the time instances when the truck enters the scan zone of each beam since the distance between individual beams is known. Likewise, as the truck is departing the detection zone, the last two or more consecutive frames can be utilized in the same manner to estimate another speed. As long as the vehicle is not occupying the entire set of 16-beams, data from such frames can be used similarly to determine entry and exit speeds. The steps implemented to estimate the speed will be explained in the next sub-sections.

4.2.2.1 *Using a Constant Speed Value (CS)*

In the preprocessing of the LiDAR data, each cloud point's height is computed with respect to the roadway surface. First, we need to remove any points less than 10". In this way, we can observe only vehicle cloud points. Second, whenever a vehicle is observed in the FOV, the laser id is recorded where the maximum LiDAR point belongs until all the beams are occupied. Then we can apply Equation (2) for estimating the speed for each consecutive frame.

$$\tilde{u}_i = \frac{X_{i+1} - X_i}{T_{i+1} - T_i} * 10^6 * c \quad (2)$$

where:

- \tilde{u}_i = estimated speed in any given two-consecutive frames i and $i+1$
- X = x coordinate in 3D cloud points
- T = Timestamp
- i = Frame number
- 10^6 = Conversion from microsecond to second
- c = conversion factor, e.g., to convert from meter per second (m/s) to mile per hour (mph)

When we compute each \tilde{u} n times, we can average estimated speeds using Equation (3).

$$\tilde{u}_{ave} = \frac{1}{n} \sum \tilde{u}_i \quad (3)$$

For example, Figure 17 shows projected 2D cloud points for a truck progressing through the FOV. Speeds can be estimated by frame numbers 8221-8222 for entering and 8231-8232 for departing. These speeds are then averaged to find a constant average speed for the vehicle. It should be noted that the precision of this method is limited since the distance can only be measured in increments of the distance between two consecutive beams. For the installation, this increment is about one foot. Given the fact that the time between two frames is 0.1 seconds, this discretized measurement of travel distance translates to approximately ± 7 mph maximum error (worst case) for a truck traveling around 50 mph. However, since multiple estimates are utilized, the actual error is expected to be lower than this. For example, in Figure 18, green points are the estimated speeds from entry and exit frames, and blue points are the averaged constant speed value across the frame numbers.



Figure 17. A 2D view of truck generated from each frame as it progresses through FOV (colors represent reflectivity)

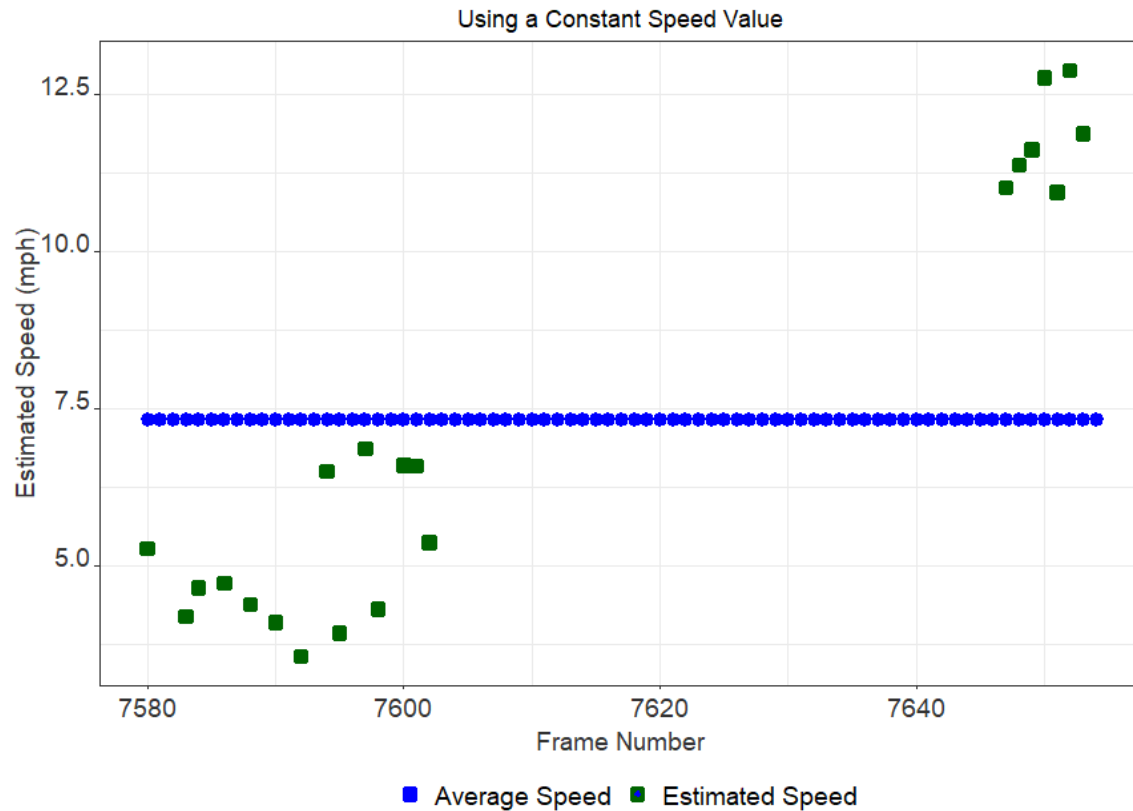


Figure 18. Estimated and averaged distributed speed across the frames

4.2.2.2 Speed estimation by Interpolation (IS)

Speed estimation is challenging under congested and stop-and-go traffic conditions because speed is different when a truck is entering into and exiting from the LiDAR area or when the truck is idling. The calculated constant average speed may cause misalignment at the vehicle profile construction. Thus, merged LiDAR points may not represent the actual truck profile.

Since we know the entry and exit speeds, we can first interpolate, then apply a smoothing method to estimate speed values for each frame. In Figure 19, green square points are the

estimated speeds from entry and exit frames, and blue circle points are the interpolated speeds that are calculated from entry and exit speeds.

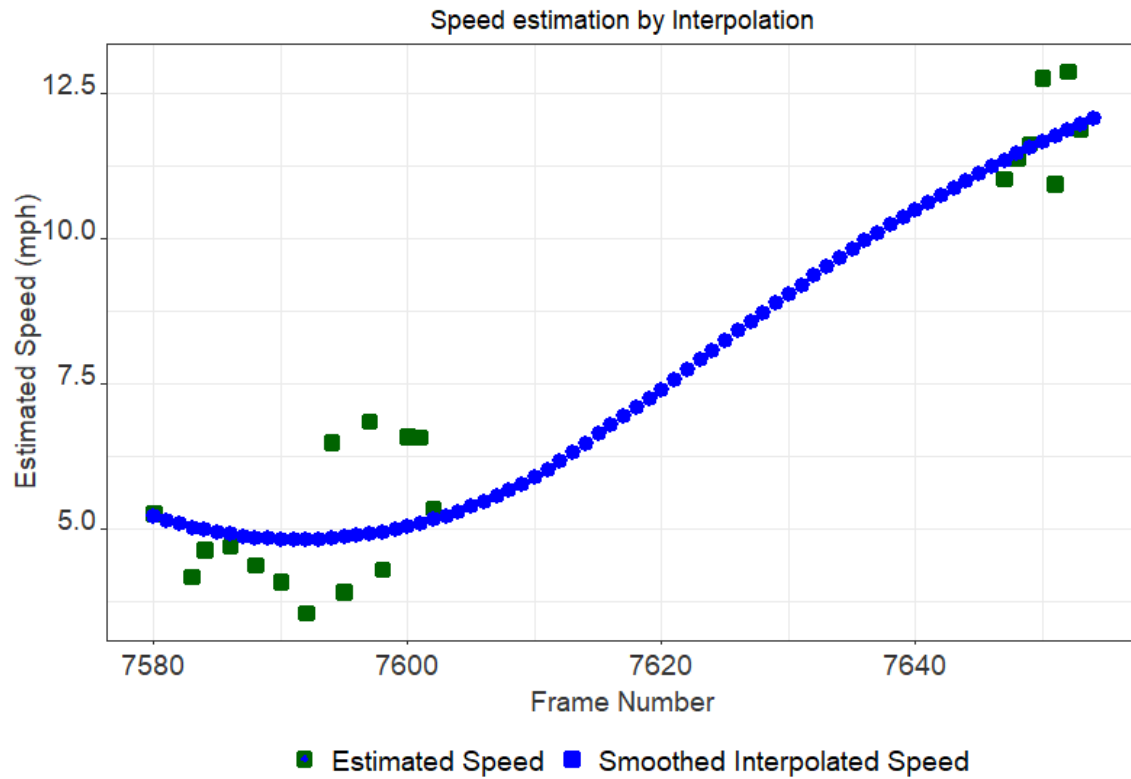


Figure 19. Estimated and smoothed interpolated speed across the frames

4.2.2.3 Speed Estimation by Tracking Axles

The LiDAR points returned from the vehicle body are used for CS and IS speed estimation. However, there is still a possibility to change speed while in the LiDAR FOV in congested traffic. In this condition, CS and IS methods may not work. As a result, vehicle profile construction could be erroneous. To estimate speed in that condition, we can use LiDAR points that are returning from axles. Once we identify the axle trajectories in the FOV, we can estimate

the speed of a vehicle. Then, we can update the speed values for the frames when axles are in the FOV. However, tracking the axles from LiDAR cloud points requires additional filtering and processing of the data. These processes will be explained in detail in the next sub-sections.

4.2.2.3.1 Identifying Axles Area

To track the trajectory of the vehicle axles, we must use the longitudinal direction of the FOV, which is “x” direction and time instances from the transformed LiDAR cloud points. First, we need to extract only cloud points returning from the region of interest which is the axle area. Since the height of each cloud point was computed previously, we can filter out the rest of the cloud points using the thresholds. The top threshold can be set at 12 inches from the pavement to remove the cloud points returning from the vehicle body. Likewise, we also need to remove cloud points that are returning from the roadway, for which a threshold is set at 2 inches from the roadway. In other words, we get the cloud points between two inches and twelve inches from the roadway surface. The red cloud points in Figure 20 are the visual representation of the selected region of interest for an FHWA Class 9 semi-truck hauling a dry-van trailer. This process is applied to every frame on the vehicle cloud points.

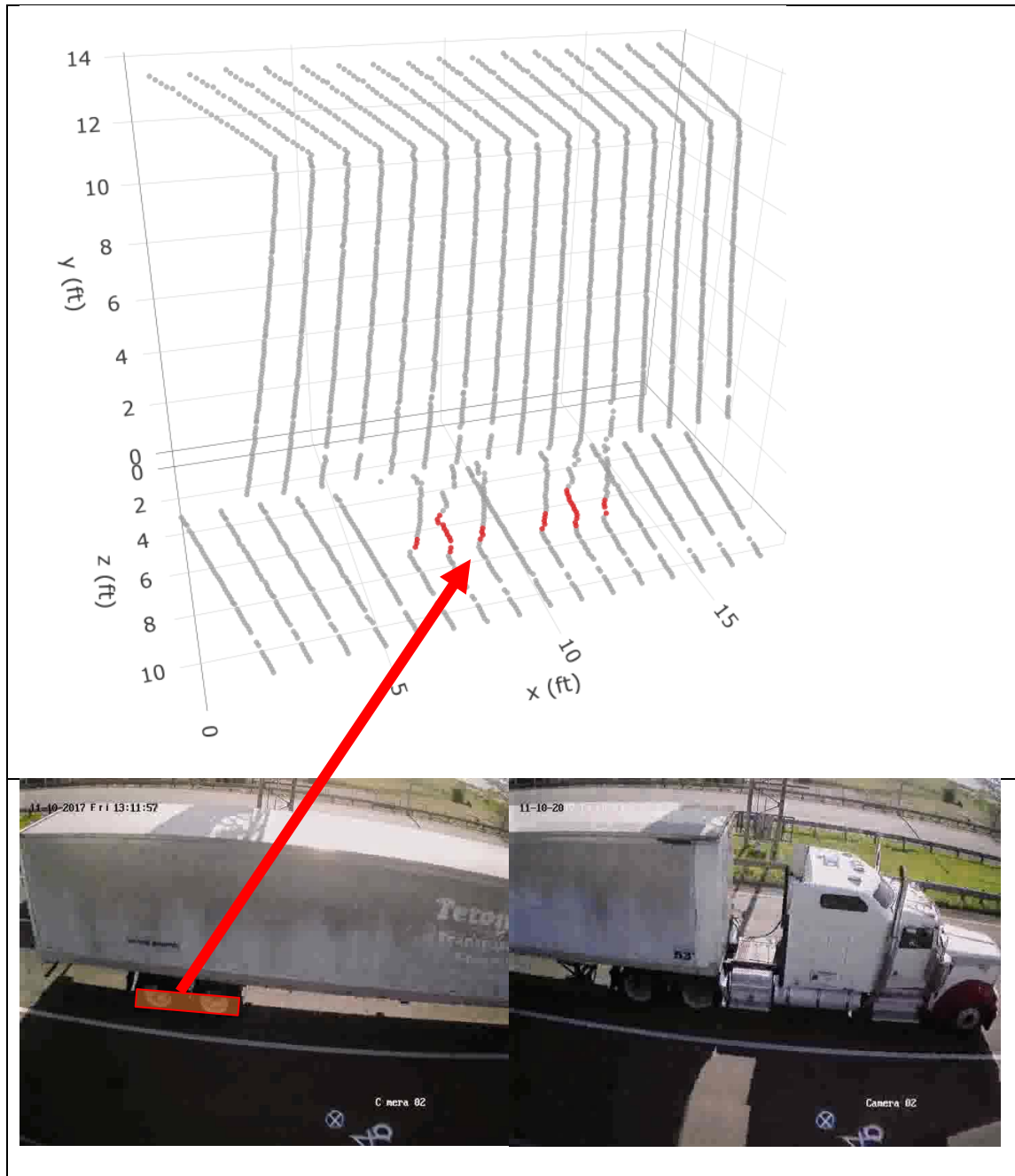


Figure 20. Used cloud points for the axle trajectory in the region of interest for FHWA Class 9 semi-truck hauling a dry-van trailer

4.2.2.3.2 Counting Number of Cloud Points

Sometimes, there are devices on the vehicle or body frame that are close to the pavement. Cloud points from these places may also fall into a defined threshold and can be identified as part of the axles or tires. The number of points from these parts is less than the total number of cloud points returning from the tires. When we count the number of points at a given time and location at the LiDAR FOV, density can vary where the tires are located. For example, in Figure 21, all the data points are mapped to a pre-defined grid. On the grid, the x-axis (time) represents time and the y-axis (direction) represents trajectory of the axles. The higher number of points, which is darker in Figure 21, indicates the higher possibility of being an axle.

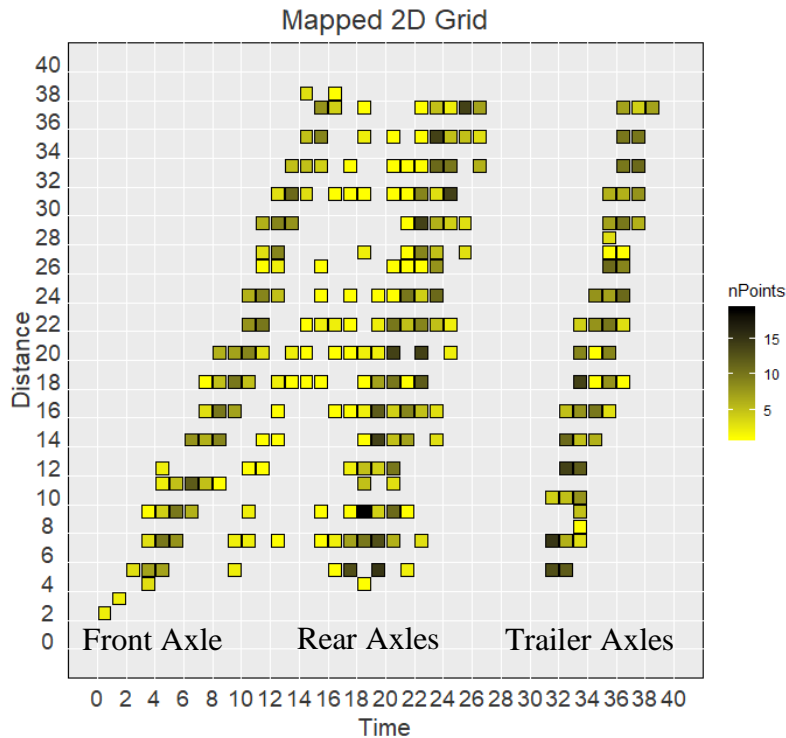


Figure 21. Axles trajectory for an FHWA Class 9 semi-truck hauling a dry-van trailer

Since we created a trajectory plot from cloud point data, the slope from each axle provides the speed of that region. If we find the slope for each axle, we can update the speed of the vehicle for given frames, and this could improve the speed estimation.

On the other hand, each truck produces a different trajectory due to different speeds and some other LiDAR points returned from the body. These kinds of samples increase the complexity of trajectory creation. In other words, each truck has a unique trajectory; therefore, we can't simply create one model for every truck. However, in the image processing and computer vision literature, automatic detection of lines is a classic problem that we can utilize to develop a speed estimation method. The proposed improved speed estimation algorithm based on the automatic line detection method is explained in the next section.

4.2.2.3.3 Hough Transform

Hough Transform (HT) [46, 47] is an effective method for detecting straight lines in the images, even in the presence of noise and missing data. We can develop a method using the HT to estimate the best line in the axle trajectories.

First, we need to prepare an empty 2D grid to map axle trajectories. The identified axle trajectories for the vehicle will be mapped onto that 2D grid space. While transforming the trajectories to the pre-defined grid, the number of cloud points for each cell will also be recorded for the voting. Consider the slope-intercept equation of a line, which is " $y = ax + b$ ". The x and y values are known from the grid; a is the slope we are looking for. From the given number of frames of the collected LiDAR cloud points, we can roughly estimate the speed and set a range for the possible slope candidates. In other words, we can quantize the parameter space as $P[a_{min}, \dots, a_{max}][b_{min}, \dots, b_{max}]$. Then using the HT method, we can find $b = -xa + y$, which is y-intercept in the coordinate system using the algorithm in Table 5.

Table 5. Pseudocode for Hough Transform line detection

```

For each time and space (x,y)

For( $a = a_{min}; a \leq a_{max}; a++$ ){

     $b = -xa + y$ ; #round off if needed

    ( $P[a][b]$ ) ++; #voting

}

Find local maxima in  $P[a][b]$ 

```

The maximum number of voted slopes are the possible lines that we are interested in. As seen in Figure 22, lines on the grid are drawn based on the HT voting method. Lines are passing on the grids where the maximum number of cloud points are. It is expected that the lines form clusters. The next step is finding the position of these clusters. It can be found by using the K-means clustering algorithm, which will be explained in the next section.

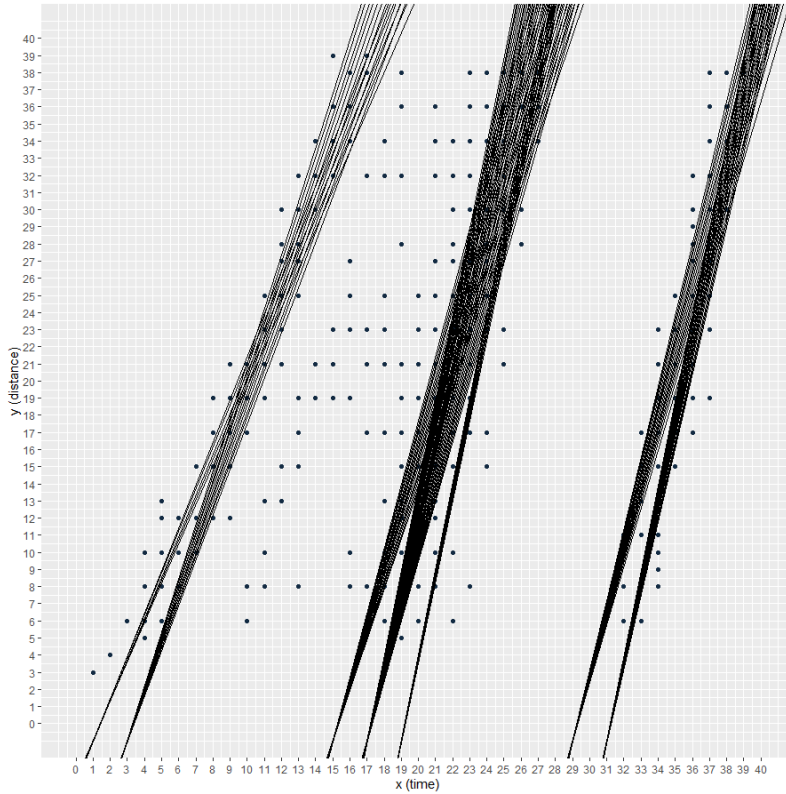


Figure 22. First 100 lines based on the HT voting method

4.2.2.3.4 K-Means Clustering

K-means clustering is an unsupervised machine learning algorithm that iteratively measures the distance from the centroids to each data point. The “K” is the identifier for a cluster size of a given data. One can find the best optimum k-value iteratively from the given data set. However, in this research, we defined the $k=3$, which represents three groups of axles, namely:

- 1) Front axles, 2) Rear axles, and 3) Trailer axles.

The input for the k-means cluster is the x-intercept of the slope, which is time. After we identify the clusters, the most voted line is selected from each cluster, as seen in Figure 23. Then, we can locate the frame numbers with the slope value, which is the speed of the vehicle at that

time. The estimated speed values, found by using tracking axes, can be seen in red in Figure 24.

Estimated speed for each method can be seen in Figure 25.

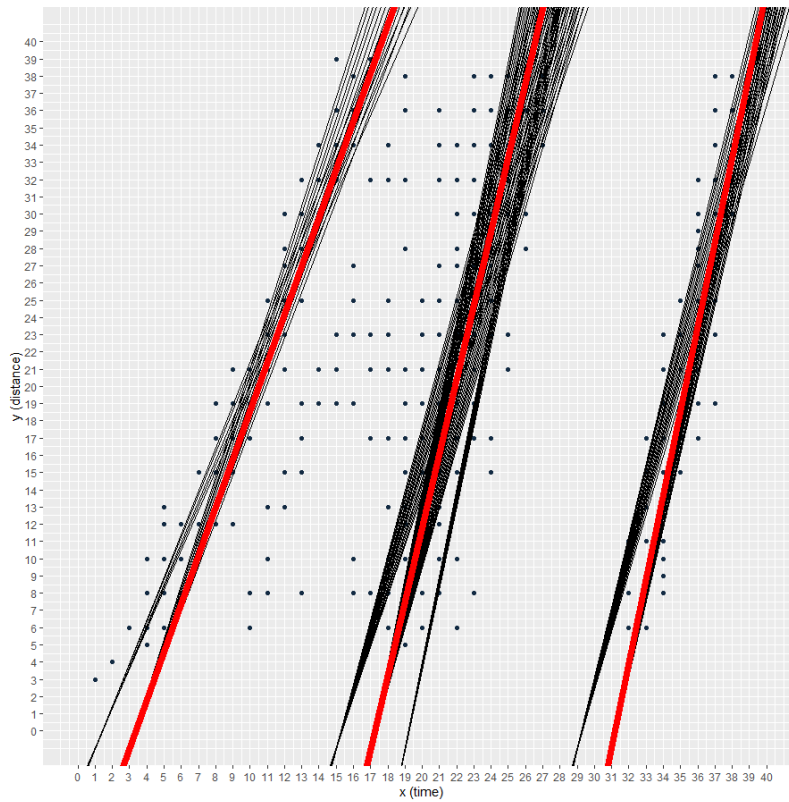


Figure 23. K-means identified dominant lines in the clusters

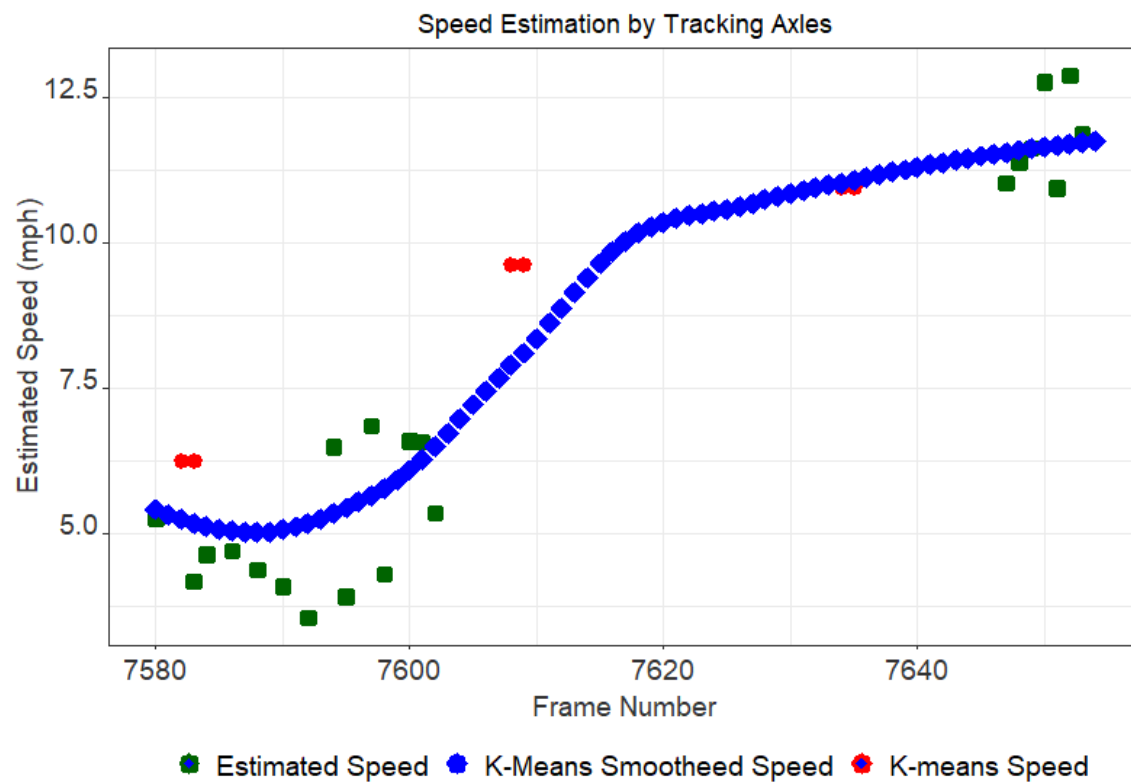


Figure 24. Estimated, k-means, and smoothed speed across the frame numbers

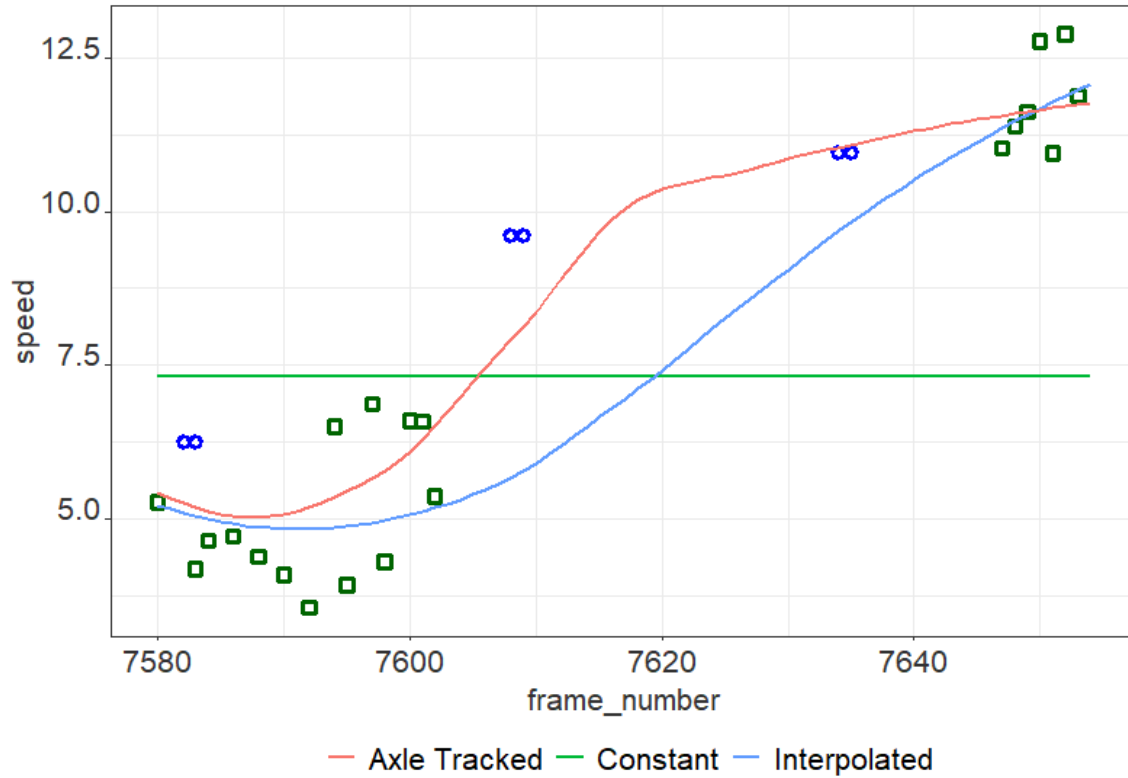


Figure 25. Speed values for reconstructing the vehicles profiles

4.2.3 Constructing Vehicle Profile

Based on the estimated speed, the frames belonging to the same trucks are then merged by shifting the consecutive frames accordingly. Since all the frames are merged, the 3D LiDAR points can be projected onto voxel grids. The size of the voxels can be adjusted based on the application. In this research, the voxels are set to be one cubic inch (1 in^3). It is also possible that LiDAR points can be projected onto 2D grids. Projecting the data to the 2D grid reduces the computation load needed to process this large number of points per truck in the feature extraction. For example, a typical FHWA Class 9 truck spends about 1-2 seconds within the LiDAR detection zone in free-flow speed. Within this time, LiDAR generates around 30,000

points. This projection reduces the data points drastically to around 1,000 points. A constructed sample of 2D and 3D truck profiles for each method can be seen in Figure 26 and Figure 27, respectively. The vertical line in Figure 26 is the 53 feet line. The trailer in this plot is 53 feet.

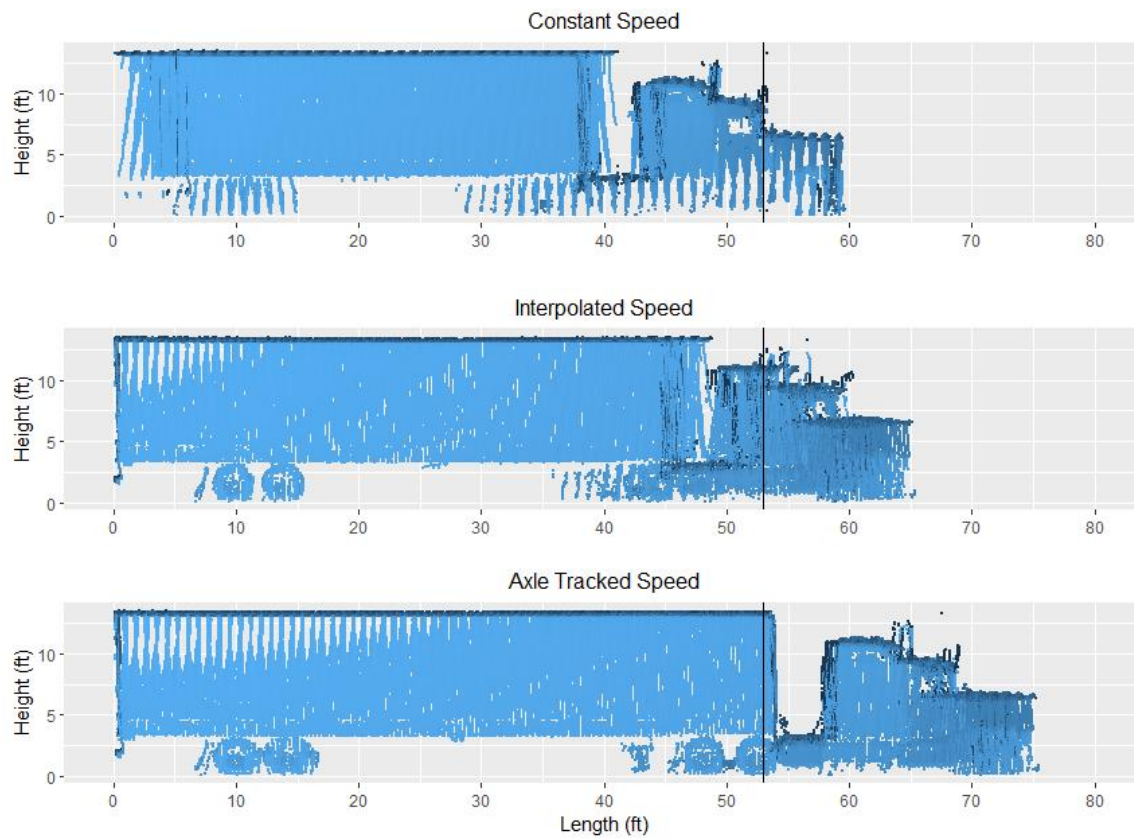


Figure 26. Constructed 2D truck profiles using proposed three-speed estimation methods

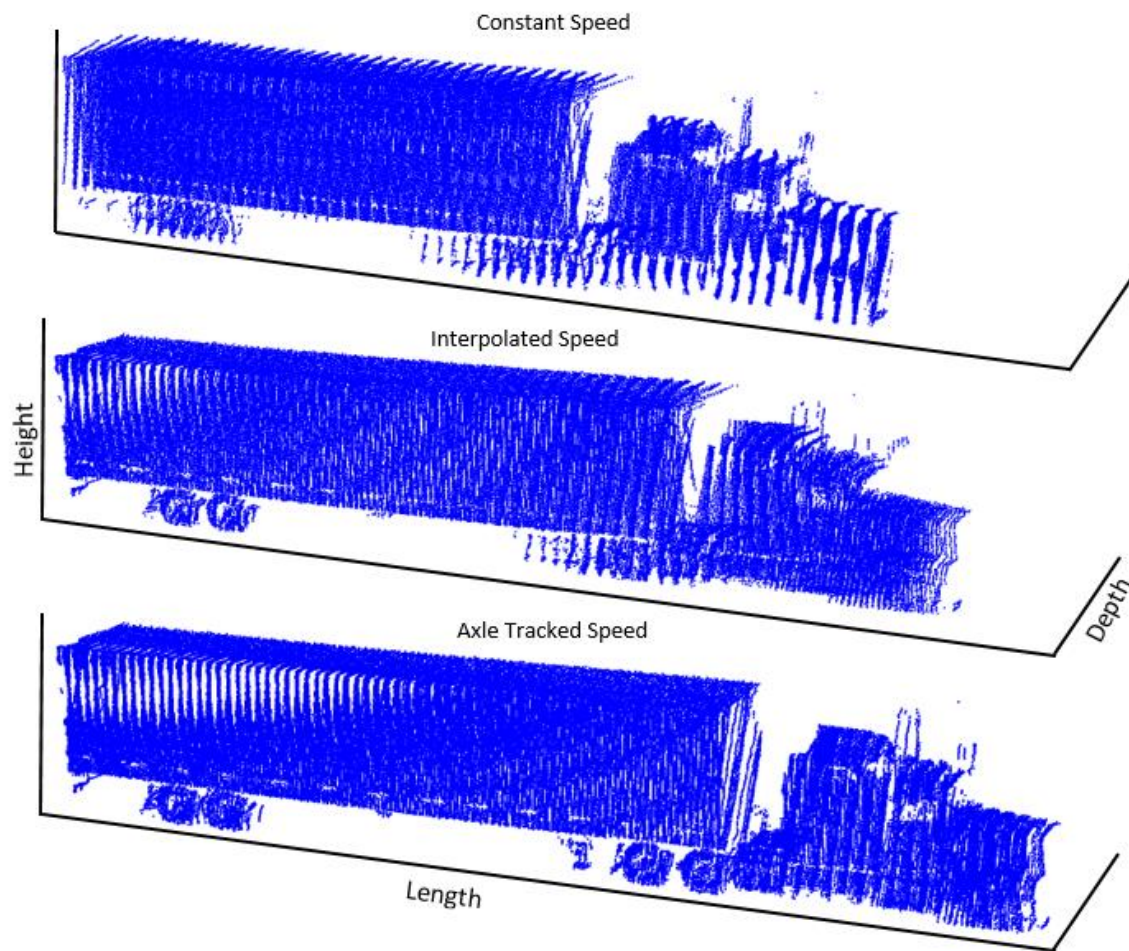


Figure 27. Constructed 3D truck profiles using proposed three-speed estimation methods

4.2.4 Speed Estimation Methods Analysis

As mentioned earlier, at the data collection site, there is no speed measurement instrument available. In practice, when the speed estimation is accurate, less error is expected in the vehicle profile construction from the LiDAR cloud points. In order to test speed accuracies estimated from CS, IS, and AS based vehicle profile construction, we can measure trailers with known lengths. For example, there are four different lengths of intermodal containers: 20, 40, 45, and 53 ft. We can measure these trailers and check the estimation error rate.

Table 6 shows the sample size for each type of intermodal trailer length. The 40 ft intermodal container in the dataset has the largest sample. It is sufficient to use 40 ft containers for testing the speed estimation methods.

Table 6. Intermodal containers sample size

Intermodal Container Length	Sample Size
20 ft	303
40 ft	1,919
45 ft	46
53 ft	32

A method for estimating the length of the trailer will be explained in Chapter 4, section 4.1.1. As can be seen in Figure 28, there is a correlation in the speed and number of frames. When the speed is decreasing, the number of frames is also increasing. In the dataset, 73% of the samples have fewer than 15 frames, 22% have between 15 and 29 frames, and 6% have greater than or equal to 30 frames (see Figure 29). Table 7 shows the statistics for the number frames in each bin.

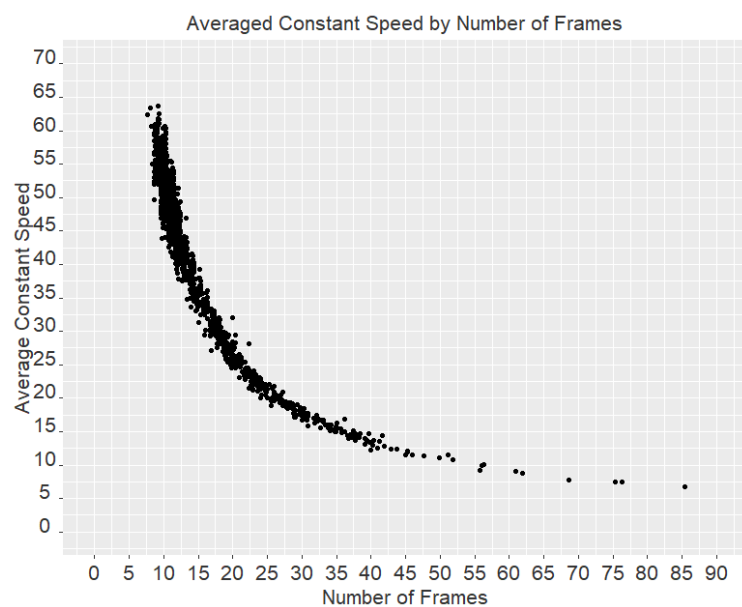


Figure 28. Averaged constant speed by frame number for 40 ft intermodal containers

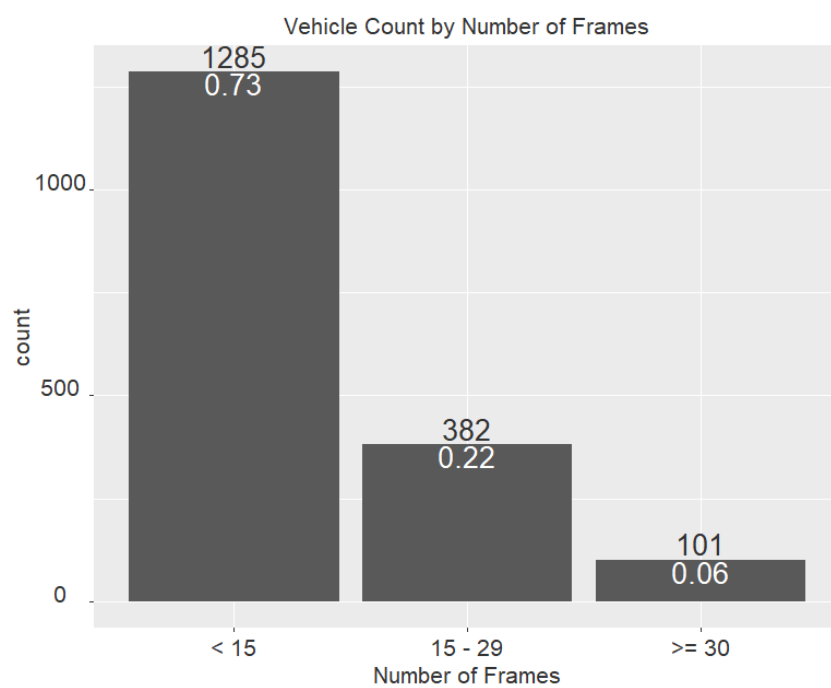


Figure 29. 40 ft Intermodal container count by the number of frames into three bins

Table 7. Statistic of estimated speed and intermodal container length by three-speed estimation methods

Number of Frames	Min	Max	Mean	Median
	Averaged Constant Speed (mph)			
< 15	33.66	63.75	49.15	49.82
15 - 29	17.12	39.26	27.35	27.46
>= 30	6.77	18.59	14.76	15.24
	Length in Constant Speed (ft)			
< 15	31.50	52.82	38.62	38.71
15 - 29	35.43	44.62	38.94	38.88
>= 30	30.51	45.28	39.00	39.21
	Length in Interpolated Speed (ft)			
< 15	31.50	53.97	38.73	38.71
15 - 29	35.27	43.80	38.81	38.71
>= 30	34.45	45.77	39.30	39.21
	Length in Axle Tracked Speed (ft)			
< 15	31.50	52.49	38.41	38.55
15 - 29	34.28	43.31	38.38	38.55
>= 30	30.18	44.13	38.53	38.39

The estimated length distribution for each speed estimation method can be seen in Figure 30 for the number of frames less than 30 and Figure 31 for the greater than or equal to 30 frames. As can be seen in the box-plot in Figure 30, three of the speed estimation methods perform similarly at high speeds or when the number of frames is less than 30 because there is no speed change in the LiDAR FOV. There are some outliers due to distance within the beam, which was explained previously in section 3.2.2. However, in Figure 31, one can see that axle tracked speed estimation fixed some of the distorted vehicle profiles. For example, significant improvement can be seen in Figure 26, which is vehicle profile construction by axle speed estimation. On the other hand, the estimated length has more variance compared to constant and interpolated speed estimation methods. The variance can be reduced by adjusting the minimum and maximum

speed in the search space for the axle tracked speed estimation method and by developing a more robust technique. This is left for future research.

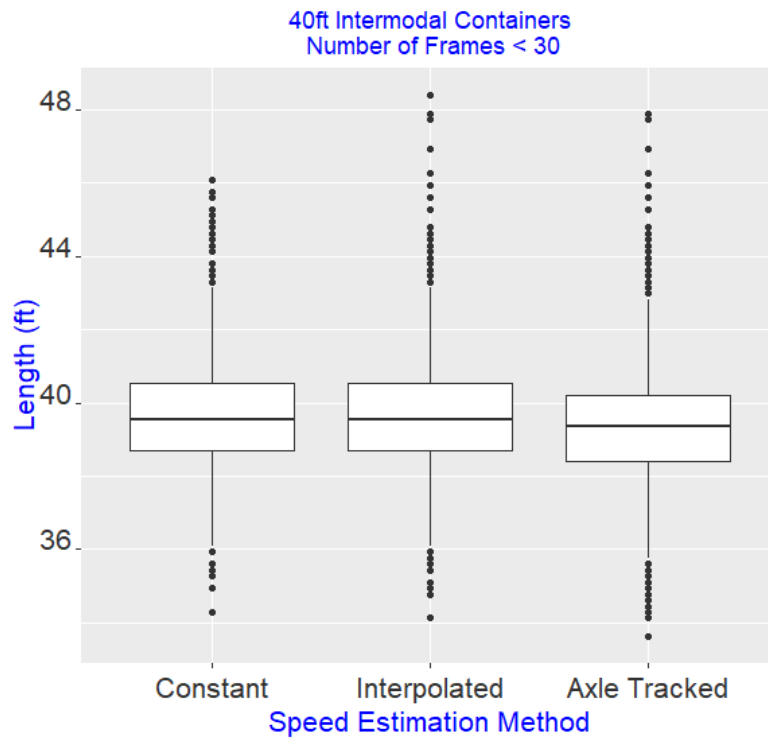


Figure 30. Speed estimation methods tested on 40 ft intermodal containers. The number of LiDAR frames is less than 30

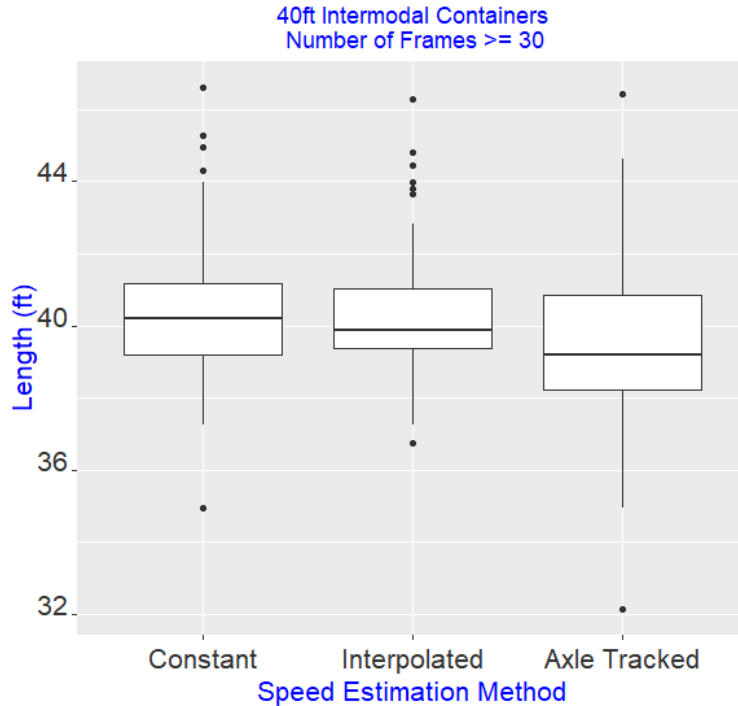


Figure 31. Speed estimation methods tested on 40 ft intermodal containers. The number of LiDAR frames greater and equal to 30

4.3 Classification Algorithms

In the literature, there are a wide variety of classification methods available [48]. In this research, the following commonly used supervised machine learning algorithms are implemented with the k-fold cross-validation ($k = 10$) to prevent bias within the dataset:

1. K-Nearest Neighbors (KNN) is a non-parametric approach and commonly used classification method due to its simplicity [49]. This algorithm has only one parameter, K, the number of neighbors. The optimum K value is selected based on the best accuracy from the training phase.
2. Multilayer Perceptron (MLP) is a feed-forward neural network structure that consists of input, hidden, and output layers [50]. In this method, the input layers are the

- features, and the output layers are classes. The number of neurons in the hidden layer is optimized based on the best output accuracy in the training dataset.
3. Adaptive Boosting (AdaBoost) is repeatedly fitting a sequence of weak learners (such as small decision trees) to data and produces a final prediction through weighted sum [51]. Grid search is implemented in order to find the optimum combination of boosting object (mfinal) and maximum depth (maxdepth) to be used in the prediction.
 4. Support Vector Machines (SVM) solves an optimization problem on training data to find the optimal hyperplane to separate classes [52]. In this method, a Gaussian Kernel and soft margin penalty, called box constraint, is applied. Each combination of a box constraint and a kernel scale make up a single SVM model. Grid search is implemented in order to find the optimum combination of box constraint and kernel scale parameters in the training phase. The SVM model with the best parameters is then used on the test data to predict the trailer type.

CHAPTER 5

TRUCK-TRAILER CLASSIFICATION

This chapter presents the trailer classification algorithms and results. These findings were recently published in the Journal of Intelligent Transportation Systems [41].

5.1 Feature Extraction

Each truck trailer has its own shape, characteristics, and dimensions. To be able to categorize trucks into distinct groups, pertinent features that help distinguish trucks in each group are needed. All features are based on cloud points within certain regions of interest defined over the 3D-truck profile shown in Figure 32. Overall, 18 features are computed.

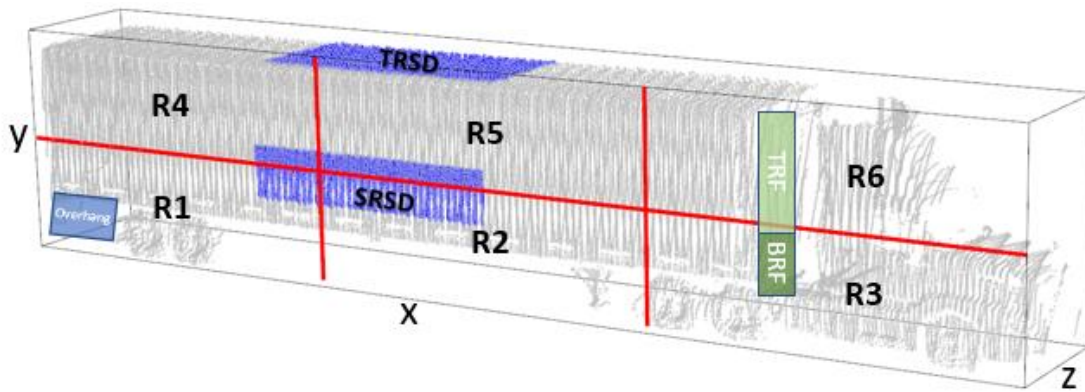


Figure 32. Extracted features annotated on a 3D profile of a refrigerated dry van

There are 11 regions of interest shown in Figure 32. Six of them are simple rectangular cells (R1, R2, ..., R6) to cover the entire truck and are found by dividing a given 2D-truck-profile into six equal-sized rectangular cells. This is accomplished by simply dividing a 2D-profile to create two rows and three columns, as shown in Figure 32. The horizontal dimension

of the rectangular regions is fixed at 7 ft since the truck height is almost always below 14 ft. The vertical dimension is set to one-third of the calculated truck length to evenly split the total length into three equal pieces. It should be mentioned that after all data points are placed into the 2D-grid, the truck length measurement becomes relatively straightforward. The overall truck length and height are found by simply measuring the distance between extreme x- and y-coordinates, respectively.

The remaining five regions, labeled as Overhang, Top Reefer (TRF), Bottom Reefer (BRF), Top Region Standard Deviation (TRSD), and Side Region Standard Deviation (SRSD) are selected to capture specific data to be able to distinguish among the box-type trailers (e.g., dry vans and intermodal containers and their reefer counterparts). For example, the density of cloud points within the Top Reefer area (TRF) and Bottom Reefer area (BRF) regions will be valuable for detecting whether a trailer has a refrigeration unit. These are placed at the front of the trailer where the refrigeration unit is expected. The other two features, namely “TRSD” and “SRSD” regions, are important for distinguishing between box type and other trailers (e.g., platforms, auto transporters, and tanks). In order to locate these regions, the trailer dimensions need to be estimated. This is explained in the next subsection, which is followed by additional subsections describing the extracted features.

5.1.1 Trailer Length and Height

Trailer length and height are not used as input features in the classification models. However, these two are needed to define the four regions of interest: “TRF,” “BRF,” “TRSD,” and “SRSD.” These four regions are particularly important for identifying trailer type, as mentioned above. To determine the x-coordinate for the frontend of the trailer, a rectangular region (e.g., 2 ft high and 6 ft long) much smaller than the trailer is selected from the middle of

the trailer (Figure 33-a). As indicated earlier, each grid cell contains the average z -coordinate (depth) of all LiDAR points corresponding (i.e., projected) to that cell. We then collapse the vertical dimension of this sample (2 ft by 6 ft) rectangle by taking the average of all z -coordinates for all y -levels at a given x -coordinate (Figure 33-b). This results in a one-dimensional vector (along the x -axis) with z -coordinates as the variables. We then fit a simple linear regression model of the form $z = mx + b$ to this vector (Figure 33-c). We then use this model to predict the z -coordinate (depth) as a function of x , where x is now extended further towards the tractor unit where the frontend of the trailer is expected to be. By measuring the difference between predicted z (depth) and actual z (depth), we can identify a sudden drop or change in the surface depth (Figure 33-d). The x -coordinate where the first sudden drop (seen as a red star in Figure 33-d) occurs will be identified as the frontend of the trailer. This technique is repeated to determine the backend, top, and bottom of the trailer as well by scanning in the relevant direction. Once these four coordinates are found, the box type trailer dimensions are straightforward to compute.

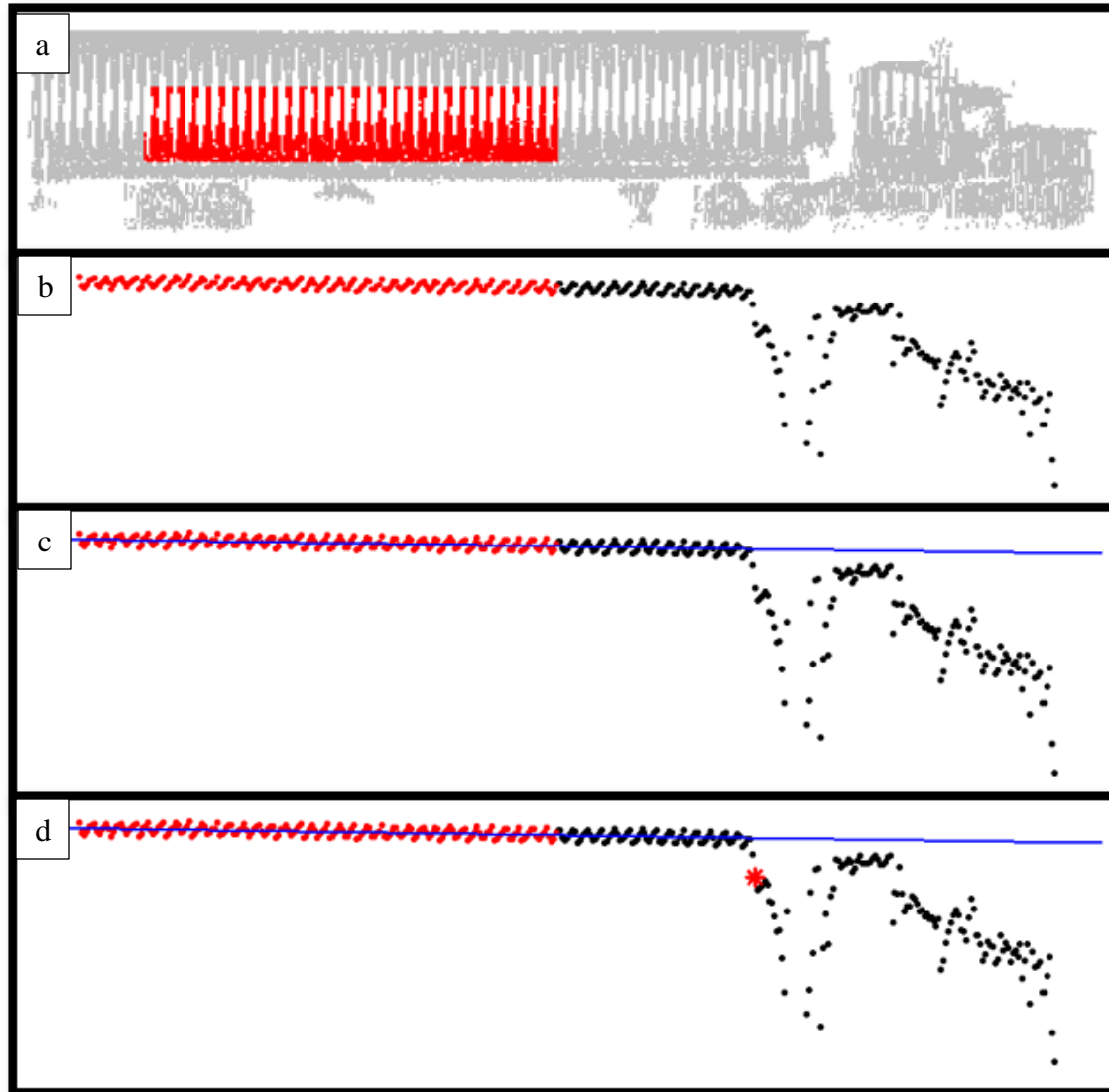


Figure 33. Trailer length and height calculation steps

5.1.2 Standard Deviation of Residuals (SDR) and Relative Density (RD)

In order to distinguish each one of the ten types of trailers considered here, we propose to divide a given 2D-truck-profile into six equal-sized rectangular cells (R1, R2, etc.). For each one of these cells, we extract raw LiDAR point cloud data contained within the boundaries of each cell and then fit a linear model of the form $\hat{z} = \alpha x + \beta y + c$, where x is the horizontal coordinate,

y vertical, \hat{z} predicted depth (see Figure 15-b), and α , β , and c are model parameters. We then compute residuals, i.e., $z - \hat{z}$, for each point contained within the boundaries of each cell. For each one of the six cells, we compute the Standard Deviation of these Residuals (SDR). In addition, for each cell, we count the number of points and divide that by the total number of points for the truck. This gives the Relative Density (RD) of points in that cell as compared to the total number of points. Since there are six cells and two variables (SDR and RD), this will produce 12 variables as input to the machine learning methods for classification.

These two types of variables exhibit enough variation among the trailer types to allow accurate classification, as shown later in the paper. For example, the surface smoothness of box-type trailers is quite different from other types of trailers, such as platforms, tanks, and specialty trailers. The VLP-16 LiDAR sensor is accurate enough to detect the small depth variations on the side surface of these trailers – enough to distinguish between containers and dry vans.

5.1.3 Top Reefer Density (TRD) and Bottom Reefer Density (BRD)

Dry vans and refrigerated dry vans look the same, except the latter has a refrigerator unit attached to the dry van's frontend. The refrigerator unit does not cover the whole space on the frontend. There is a gap between the tractor chassis and underneath the refrigerator unit, as seen in Figure 32. Therefore, in the 2D profile of a refrigerated unit, there are more LiDAR points observed at the top section protruding forward from the trailer. This information can be captured by defining two regions (rectangles) at the frontend of the trailer. Since the x-coordinate of the frontend is estimated, we can go a certain distance towards the tractor (about 2 ft) from the trailer edge (see the highlighted sections labeled as TRF and BRF in Figure 32) and calculate the density of points in a rectangle close to the top of the trailer and another rectangle at the bottom of the trailer. The heights of the bottom and top rectangles are taken as 2 ft and 4 ft, respectively.

We then simply count the number of grid cells (i.e., two by two inches cells used in 2D projection) that are not empty and divide the count by the area of the rectangle to find the density of points.

5.1.4 Side and Top Region Standard Deviation (SRSD and TRSD)

Since we have the trailer position in the cloud point, extracting the side and top regions is straightforward. We simply locate rectangular regions at the center point of the trailer for extracting such data. For the side region, a 3.5 ft by 16 ft rectangle area is extracted, shown in blue at the side of the trailer labeled SSD in Figure 32. For the top region, this rectangle is three-fourths of a trailer, the width by 16 ft, shown in blue at the top of the trailer labeled TSD in Figure 32. The cloud point data gathered for these regions are extracted, and the same technique as above is applied to find the standard deviation of the residuals, which was discussed earlier. If there is no point available in the selected regions (e.g., in the case of open-top vans), its value is set to “999”.

5.1.5 Middle Height (MH)

Similar types of trailers are expected to have similar heights and dimensions. For example, the height of box-type trailers is between 10 ft and 14 ft from the ground whereas the heights of platform trailers vary based on the commodity carried. In any case, the trailer height does not always help in clustering all trailer types but certainly would be a useful feature for more standard types such as containers and dry vans. Rather than simply using the maximum height, the trailer height is calculated at the midpoint of the trailer. Since the trailer dimensions are already determined from the steps discussed above, the middle height (MH) is simply computed from the highest LiDAR points located at the midpoint of the trailer along its length.

5.1.6 Overhang (OH)

Another useful variable is the overhang distance or the distance from the trailer end to the point where the last rear tire is. This distance is quite small for intermodal trailer containers where it is typically larger for enclosed or dry vans. Rather than attempting to measure the overhang distance from the 2D profiles, we use a 2 ft high by 4 ft long rectangle starting at the origin of the coordinate system and measure the density of points within this rectangle. This turns out to be a good surrogate variable to capture the variation in rear overhang among the trailers of different types.

5.2 Classification

Overall, 18 variables listed below are found to be effective for the purposes of this study and are used as input features for the classification methods listed above. Their descriptive statistics can be seen in Table 8.

- Standard Deviation of Residuals (SDR) for each region one of the six rectangular regions denoted as $SDR_{R1} \dots SDR_{R6}$
- Relative density (RD) for each region denoted as $RD_{R1} \dots RD_{R6}$
- Top Reefer Density (TRD)
- Bottom Reefer Density (BRD)
- Side Region Standard Deviation (SRSD)
- Top Region Standard Deviation (TRSD)
- Middle Height (MH)
- Overhang (OH)

These variables show enough variation across the different truck trailers. As seen in Figure 34, all box-type trailers have residual standard deviation between 0.5 inches to 3.0 inches, because there is less variability on these surfaces. Since other trailers such as automobile transport and platform trailers have no uniform shape, their residual standard deviations are much larger. On the other hand, tank trailers are also differentiated based on the TRSD variable where the majority of them 0.5 ft or above.

Table 8. Features extracted for each truck and descriptive statistics

Feature	Abbreviation	Unit	Min	Max	Mean	1st Qu.	3rd Qu.
St. Dev. Residuals Region #1	SDR _{R1}	Feet	0.166	3.297	1.055	0.622	1.215
#2	SDR _{R2}	Feet	0.071	3.406	0.797	0.254	1.202
#3	SDR _{R3}	Feet	0.649	2.631	1.627	1.511	1.753
#4	SDR _{R4}	Feet	0.000	4.103	1.550	1.551	1.851
#5	SDR _{R5}	Feet	0.000	4.104	1.555	1.541	1.800
#6	SDR _{R6}	Feet	0.650	3.334	1.926	1.743	2.096
Relative Density Region #1	RD _{R1}	Ratio	0.047	0.422	0.146	0.109	0.147
#2	RD _{R2}	Ratio	0.035	0.392	0.136	0.099	0.142
#3	RD _{R3}	Ratio	0.061	0.483	0.182	0.148	0.197
#4	RD _{R4}	Ratio	0.000	0.392	0.208	0.210	0.257
#5	RD _{R5}	Ratio	0.000	0.358	0.211	0.206	0.258
#6	RD _{R6}	Ratio	0.006	0.317	0.116	0.098	0.138
Top Reefer Density	TRD	Ratio	0.000	0.792	0.064	0.008	0.084
Bottom Reefer Density	BRD	Ratio	0.000	0.681	0.061	0.008	0.090
Side Region Standard Deviation	SRSD	Feet	0.000	4.0210	0.437	0.118	0.186
Top Region Standard Deviation	TRSD	Feet	0.000	3.692	0.280	0.064	0.143
Middle Height	MH	Feet	1.804	14.436	12.229	12.467	13.615
Overhang	OH	Ratio	0.000	0.773	0.098	0.008	0.168

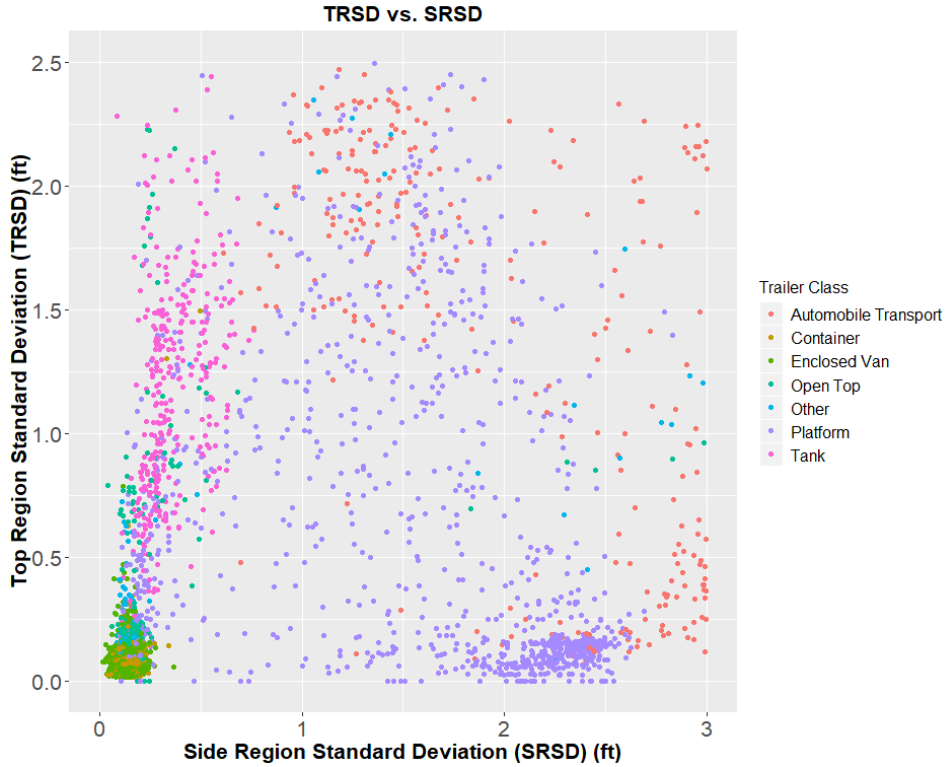


Figure 34. Residuals of standard deviation by side and top region

The ML methods are trained with the subset of the data indicated in Table 9. The training step results in determining the best model parameters to be used in prediction. All modeling work is done with the statistical programming language R [53], and Caret R-package [54] is employed for model training and development. These methods are implemented on the field data that includes over seven-thousand trucks. Based on 10-fold cross-validation applied to the training data, the model hyper-parameters that produce the best performance are found for each one of the four ML models. These parameters are namely the number of trees being 300 with a maximum depth of 10 for AdaBoost, 29 nodes for the single hidden layer MLP, 5 nearest neighbors for KNN, sigma value of 0.1, and C value of 10 for Radial Basis Function Kernel SVM. Models with these optimum hyper-parameters are then applied to the test data to evaluate

their prediction performance. The empirical analyses and results are discussed in the next section.

Table 9. Training and testing data

Trailer Type	Training	Testing	Total
20 ft Intermodal container	206	87	293
40 ft Intermodal container	1,292	553	1,845
40 ft Reefer container	164	70	234
Dry van	1,711	773	2,484
Reefer dry van	984	421	1,405
Platform	985	421	1,406
Tank	234	100	334
Automobile transport	229	97	326
Open top van/dump	110	47	157
Other (livestock, logging, etc.)	182	77	259
Total Samples	6,097	2,646	8,743

5.3 Empirical Analyses and Results

Table 10 shows the results of the ML models explained above for both the constant and axle tracked speed estimation method. The italic figures in the parentheses are the classification results based on the vehicle construction of the axle tracked speed estimation. The performance of the models is assessed by computing the Correct Classification Rate (CCR). CCR is found by dividing the number of correctly classified samples by the total number of samples in the given category.

Table 10. Results from the classification algorithms on the test data

	Proposed Method					Hernandez et al. (2016) Study	
Trailer Type	Number of test samples	CCR (Correct Classification Rate) (%)				Number of test samples	Best CCR (%)
		SVM	ADABOOST	KNN	MLP		
20 ft Intermodal container	87	96.3	97.5	96.3	96.3	13	100.0
40 ft Intermodal container	553	97.7	96.8	97.2	96.4	131	87.8
40 ft Reefer container	70	94.2	93.6	97.1	97.1	16	93.8
Dry van	773	94.3	94.1	92.3	93.6	2,329	83.5
Reefer dry van	421	91.0	91.5	91.0	92.2	1,565	75.3
Platform	421	94.9	96.0	90.4	91.8	734	87.5
Tank	100	97.1	95.1	98.0	96.1	265	80.4
Automobile Transport	97	91.1	91.1	92.1	93.1	67	91.0
Open Top Van/Dump	47	85.1	68.1	53.2	76.6	203	76.8
Other (Livestock, logging, etc.)	77	62.5	29.2	45.8	41.7	-	-
Median CCR		94.2	93.8	92.2	93.3		

In general, all tested machine learning models give a median CCR of 92% or higher in both constant and the axle tracked speed estimation method. However, the SVM model has the highest CCR among them with 94% median accuracy. Within the box-type-trailer category, the SVM model correctly identifies the trailer types with accuracies ranging from 91% (for reefer dry-van) to 98% (for 40 ft container). For the other group of classes, the CCR is 85% for open-top van/dump, 91% for automobile transport, 95% for platform trailers, and 97% for tank trailers. The “other” category has the fewest samples (only 24) in the test data, and the CCR is 63%.

Table 10 also includes results from Hernandez et al.'s (2016) study, which proposes a classification method for truck body configuration using weigh-in-motion and inductive loop signature data. Hernandez and Tok [36] classified 14 body types, which are FHWA Class 9, five-axle semi-tractors pulling single trailers. In our data, we have additional body types, but some of the classes are aggregated to maintain large-enough sample sizes. For example, we counted dropped-frame vans in the “dry van” category where Hernandez et al. (2016) have a distinct class type for them. We also aggregated 45 ft and 53 ft intermodal containers to the 40 ft container group due to having very few samples in the dataset. In our model, livestock and logging trailers are labeled as “other” types. However, Hernandez et al. (2016) have distinct classes for these. In our dataset, there are also around 40 empty container chassis available. Due to the low sample size, container chassis are counted in the “platform” category, which was done in the Hernandez et al. (2016) study as well. As shown in the table, for all body types with the exception of 20 ft containers, the proposed method in this paper gives more accurate results. This is most likely due to the richer LiDAR data and more pertinent information and features extracted from the raw data. It should be noted that Hernandez et al. (2016) considered more vehicle classes and had much larger samples in some categories.

To analyze the errors in more detail, Table 11 shows the confusion matrix (generated based on the SVM model) where the rows represent the actual class and columns of the predicted class. Clearly, most of the samples are correctly classified. However, thirty-two reefer dry-vans were predicted as a dry van; this is because of the speed variability of the truck while within the LiDAR detection zone – the methods presented here assume a constant speed. When LiDAR frames are merged, some data points don't fit correctly in the place of where a refrigerated unit is attached. Then, the model predicts that there is no refrigeration unit. This error is seen in dry

vans as well, where twenty-four of the dry vans were predicted to be reefers. Nine of the 40 ft intermodal containers were predicted as dry vans. It was found that these predictions were 53 ft containers which have characteristics similar to dry vans. For example, they also have an overhang, and the outer shell is flat material rather than the usual corrugated metal. Nine of the automobile transport trailers were predicted as platforms, which is because some auto transporters are similar to a platform when they are empty.

Table 11. Confusion matrix (for SVM method), rows represent the actual class and columns the predicted class

		↓Predicted↓									
		20ft Container	40ft Container	40ft Reefer Container	Dry Van	Reefer Dry Van	Platform	Tank	Automobile Transport	Open Top Van/Dump	Other
↓Actual↓	20ft Container	78 (96.3%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	3 (3.7%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
	40ft Container	0 (0%)	515 (97.7%)	1 (0.2%)	9 (1.7%)	0 (0%)	2 (0.4%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
	40ft Reefer Container	0 (0%)	2 (2.9%)	65 (94.2%)	0 (0%)	0 (0%)	2 (2.9%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
	Dry Van	0 (0%)	9 (1.5%)	0 (0%)	577 (94.3%)	24 (3.9%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	2 (0.3%)
	Reefer Dry Van	0 (0%)	1 (0.3%)	2 (0.5%)	32 (8.3%)	352 (91%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
	Platform	1 (0.3%)	1 (0.3%)	0 (0%)	0 (0%)	0 (0%)	357 (94.9%)	4 (1.1%)	10 (2.7%)	2 (0.5%)	1 (0.3%)
	Tank	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	2 (2%)	99 (97.1%)	0 (0%)	1 (1%)	0 (0%)
	Automobile Transport	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	9 (8.9%)	0 (0%)	92 (91.1%)	0 (0%)	0 (0%)
	Open Top Van/Dump	0 (0%)	1 (2.1%)	0 (0%)	1 (2.1%)	0 (0%)	3 (6.4%)	1 (2.1%)	0 (0%)	40 (85.1%)	1 (2.1%)
	Other	0 (0%)	1 (4.2%)	0 (0%)	2 (8.3%)	1 (4.2%)	4 (16.7%)	0 (0%)	0 (0%)	1 (4.2%)	15 (62.5%)

In order to understand if the speed is an important factor in the misclassification, we have applied Pearson's chi-squared test of comparison among different speed values. Table 12 shows the breakdown of observations and their percentages based on their estimated speed values. The speed values are divided into three different bins. Based on the freeway traffic, one can consider

the speeds less than 20 mph as stop-and-go traffic, between 20 mph and 40 mph as congested, and greater than 40 mph as free-flow. As seen in Table 12, the misclassification rate for samples collected during stop-and-go traffic is slightly greater than the other two.

Table 12. Misclassified and correctly classified truck-trailers from the SVM model disaggregated by speed

Speed (mph)	Number and % of Samples	
	Misclassified (%)	Correctly Classified (%)
<20	12 (7.50%)	148 (92.50%)
20-40	28 (5.91%)	446 (94.09%)
>40	96 (5.67%)	1,596 (94.33%)

Table 13 shows the Pearson's chi-squared test of comparison among all traffic conditions for each ML algorithm. Based on the 95% level of significance, the speed range is found to be statistically insignificant. In other words, for the sample considered in this paper, these speed levels are not found to affect truck-trailer classification accuracies significantly.

Table 13. Pearson's chi-squared tests for checking whether the classification results are affected when samples are disaggregated by speed

ML Method	p-value	Statistic (χ^2)	Statistically Significant?
SVM	0.641	0.890	No
ADABOOST	0.388	1.896	No
MLP	0.474	1.494	No
KNN	0.561	1.157	No

5.4 Summary

This chapter demonstrates how LiDAR data could be utilized to accurately predict truck trailer types. The presented methodology for processing the data involves a series of statistical models and heuristics to extract pertinent features to distinguish between different truck body types of FHWA class 9 five-axle tractor-semitrailer. Over 7,700 trucks are manually labeled, and their LiDAR data are processed. After extracting key features, various ML algorithms, including a SVM, AdaBoost, KNN, and a MLP, are trained to determine whether the subject truck is hauling a dry van, refrigerated van, 20 ft and 40 ft intermodal containers, 40 ft refrigerated container, platform, tank, car transporter, open-top van/dump, or other. The other types of trailers contain various body types, including livestock, logging, custom-designed trailer, etc. While all ML methods produce relatively high accuracies, the results of the SVM model on test data are slightly higher with a medium CCR of 94%. The results presented in this paper are compared to other studies in the literature that rely on weigh-in-motion and other sensors for truck body classification. From this comparison, it can be concluded that LiDAR provides much richer information for detecting truck body types since the presented accuracies were found to be significantly higher in general.

CHAPTER 6

DETECTING EMPTY AND LOADED PLATFORM SEMI-TRAILERS

6.1 Feature Extraction

Each truck trailer has its own shape, characteristics, and dimensions. According to the 2001 VIUS, there are 11 major body types of trucks (platform, van, auto transport, dump, grain bodies, garbage, livestock, logging, dry tank, liquid tank, and other) in the USA. When we consider these various body types, a platform type trailer does not have a unique shape when loaded. To be able to categorize truck trailers into distinct groups, pertinent features that help distinguish trailers are needed. Since we have the 3D profile of each truck, dividing the entire truck into pre-defined equal-sized voxels is expected to generate enough information for distinguishing different body types.

In order to set voxels dimensions, the longest truck and largest height and depth values are taken. In the next section, how the voxels are created is explained, which is followed by an additional subsection describing the extracted features.

6.1.1 Defining Voxels

In order to extract features, the 3D profile of the truck body is subdivided into smaller voxels. First, a large enough rectangular prism is defined such that it can enclose all trucks in the dataset. This is accomplished by simply determining the largest measurements in the coordinate directions. Measured maximum length and height of trucks are 960.0 and 178.0 inches, respectively. According to these measures, the rectangular prism is created with the size of $180" \times 144" \times 960"$ ($15' \times 12' \times 80'$), height (H), depth (D), and length (L), respectively. Then this prism is divided into equal-sized smaller voxels such that there are four in the longitudinal direction and three in lateral and vertical directions, as shown in Figure 35. This

gives a total of 36 voxels. These voxels are denoted as $V_1, V_2, V_3, \dots, V_{36}$ or V_i for the i^{th} voxel, in the remainder of this paper. These 36 “Vs” can be seen in Figure 35, where each voxel is colored. The voxels dimensions are set at $60''H \times 48''D \times 240''L$ ($5'H \times 4'D \times 20'L$). Then, for each voxel the relative density (RD) is calculated as explained in the next section.

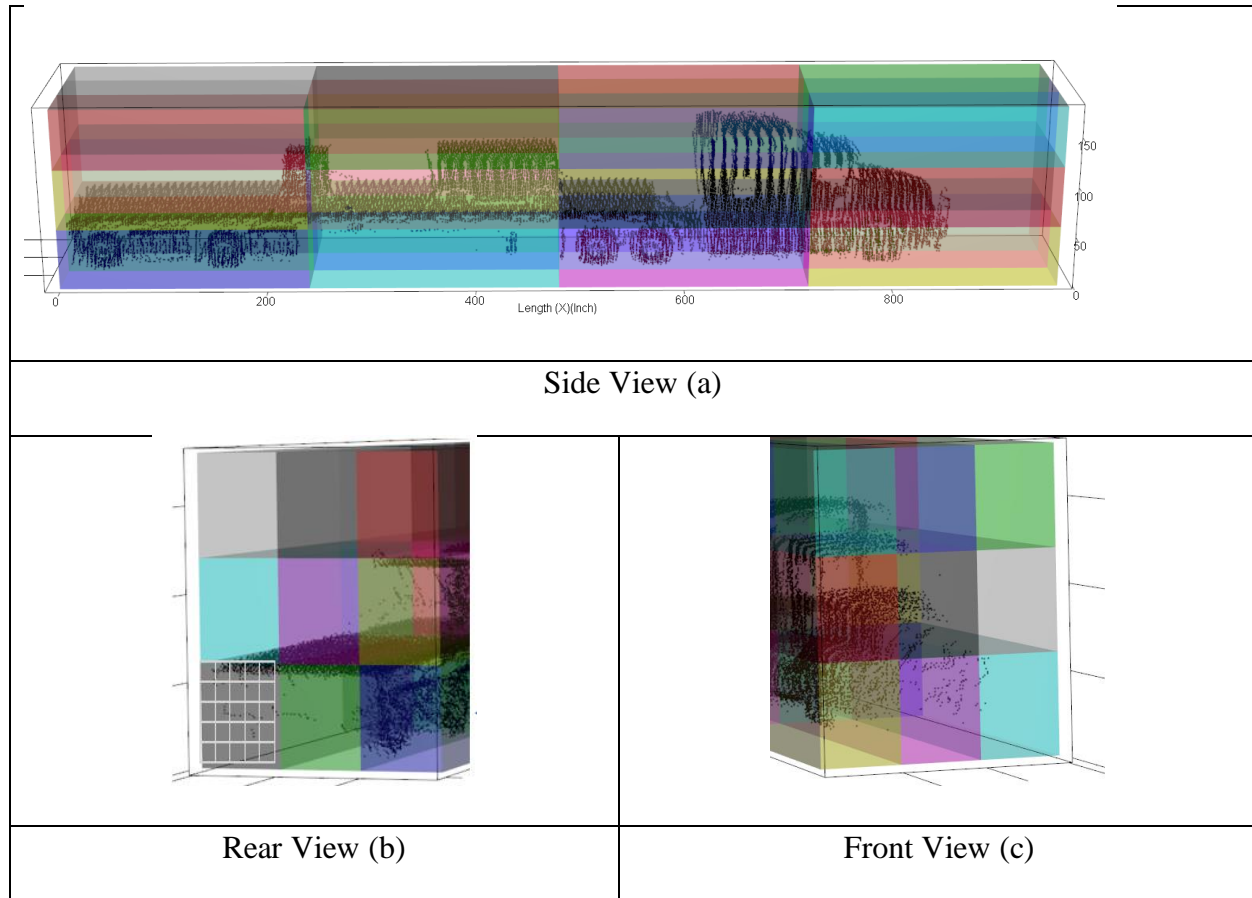


Figure 35. 36 Voxels from different views with a 3D truck cloud points. (a) Full View (b) Rear View (c) Front View

6.1.2 Relative Density (RD)

To compute the density for each V_i , first, raw 3D LiDAR data are divided into one cubic foot (1 ft^3) sub-voxels. Visual representation of these sub-voxels can be seen in the bottom left

voxel in Figure 35-b. Based on the selected dimensions for V_i 's, the total volume of each V_i is 400 ft^3 ; hence, there are 400 unit-volumes constituting each V_i . Each unit volume is considered either occupied or not depending on the presence of raw LiDAR points. If there is at least one LiDAR cloud point within a given unit-volume, it is considered occupied. The relative density (RD_i) is found by simply dividing the total number of occupied (O) unit-volumes to the total volume (i.e., 400). RD is computed for each of the 36 V_i s as shown in Equation (4).

$$RD_i = \frac{\sum O_j}{400} \begin{cases} O_j = 1 & \text{if the } j\text{-th unit voxel contains at least one LiDAR point} \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

6.1.3 Sections Height (H)

Similar types of trailers are expected to have similar heights and dimensions. For example, the height of box-type trailers is between 10 ft and 14 ft from the ground, whereas the heights of platform trailers vary based on the commodity carried. In any case, the trailer height does not always help in clustering all trailer types but certainly would be a useful feature for more standard types such as containers and dry vans. Since we already defined four sections in the longitudinal (y) direction, we simply get the highest LiDAR points (H_1, H_2, H_3, H_4) belonging to each section. Thus, we have four additional features for the classification models.

6.2 Classification

The classification step involves predicting the type of the semi-trailer based on the extracted 36 RDs and 4 Hs explained above. The truck trailers are classified into three groups: empty platform, loaded platform, and all other trailer types. To accomplish this, four classification models are created using KNN, MLP, AdaBoost, and SVM. These classification

models are well-known supervised learning techniques that have been widely used to solve classification problems in various domains. Supervised learning techniques in which labeled data are used for training and testing phases have proven to be successful in many applications. For the input, a total of 40 variables are used: relative density for each one of the 36 V_i and maximum heights measured after dividing the truck longitudinally into four equal-sections. All modeling work is done with the statistical programming language *R* [53], and *Caret R* package [54] is employed for model training and testing. Based on 10-fold cross-validation applied to the training data, the model hyper-parameters that produce the best performance are found for each one of the four ML models.

These parameters are: the number of trees being 100 with a maximum depth of 10 for AdaBoost, 37 nodes for the single hidden layer MLP, 5 nearest neighbors for KNN, and sigma value of 0.001 and C value of 1000 for Radial Basis Function Kernel in SVM. Models with these optimum hyper-parameters are then applied to the test data to evaluate their prediction performance.

6.3 Empirical Analyses and Results

Table 14 shows the results of the Machine Learning (ML) models listed above. The performance of the models is assessed by computing the CCR. CCR is found by dividing the number of correctly classified samples by the total number of samples in the given category. In general, all tested machine learning models for the empty platform provide over 92% CCR. However, the KNN model has the highest CCR among them, with 99% CCR. The CCR for the loaded platform is lower, ranging between 75% and 84%. The other types of trailers have very high CCR in all ML models.

Table 14. Results from the classification algorithms

Trailer Type	Number of training samples	Number of test samples	Proposed Method			
			CCR (Correct Classification Rate) (%)			
			SVM	ADABOOST	KNN	MLP
Empty Platform	502	215	92.1	97.2	99.1	95.3
Loaded Platform	527	225	82.7	83.1	75.1	84
Others	5,437	2330	99.3	98.9	99.2	99.3

To analyze the errors in more detail, Table 15 shows the confusion matrix where the rows represent the actual class and columns of the predicted class. As shown in the table for SVM and MLP models, 8 of the empty platforms are predicted as loaded. After closer inspection, it is found that sometimes truck drivers load their tarp onto the platform, the ML models identify the tarp as a load. The most problematic class in the dataset is loaded platform. The majority of misclassifications for the loaded platforms are predicted as the other category due to some of the loaded platforms resembling the trucks in the subcategory other. For example, some of the loads have similar shapes as a tank or container. On the other hand, the other category performed very well; only a few samples are misclassified. The major misclassified subclasses in the other category are container chassis, logging, and auto transport semi-trailers. The container chassis are sometimes misclassified as an empty platform. In the VIUS, log semi-trailers are considered as a distinct body type. They are essentially platforms with devices permanently mounted on the bed of the trailer due to its purpose of use. Therefore, we aggregated the logging trucks into the other category. The ML models misclassified these trucks as a loaded platform. This misclassification is expected because loaded log semi-trailers have similar physical

characteristics as the loaded platform. The auto transport semi-trailers have various configurations, e.g., 4-6, 7-8, 9-10 car haulers. The ML models misclassified about 4-6 car haulers as a loaded platform. Overall, the ML models distinguish the empty trailers with a 1% to 8% error rate.

Table 15. Confusion matrix for each ML model. Rows are actual, and columns are predicted values from the models for the testing data.

	Predicted			
		Empty	Loaded	Others
Actual	SVM			
	Empty	198 (92.1%)	8 (3.76%)	9 (%)
	Loaded	8 (3.6%)	186 (82.7%)	31 (%)
	Others	2 (0.1%)	14 (0.6%)	2,314 (99.3%)
	KNN			
	Empty	213 (99.1%)	0 (0.0%)	2 (0.9%)
	Loaded	23 (10.2%)	169 (75.1%)	33 (14.7%)
	Others	4 (0.2%)	14 (0.6%)	2,312 (99.3%)
	AdaBoost			
	Empty	209 (97.2%)	5 (2.3%)	1 (0.5%)
	Loaded	9 (4.0%)	187 (83.1%)	29 (12.9%)
	Others	5 (0.2%)	21 (0.9%)	2,304 (98.9%)
	MLP			
	Empty	205 (95.3%)	8 (3.7%)	2 (0.9%)
	Loaded	12 (5.3%)	189 (84.0%)	24 (10.7%)
	Others	2 (0.1%)	14 (0.6%)	2,314 (99.3%)

Note: Bold marks > 80%

6.4 Summary

In order to quantify empty trips, statistical and optimization models are developed in the literature for freight planning and analysis. In this chapter, we demonstrate how LiDAR data could be utilized to accurately predict empty platform semi-trailer trucks to support freight

analysis and planning. Over 9,200 trucks are manually labeled, and their LiDAR data are processed. After extracting key features, various ML algorithms, including a SVM, AdaBoost, KNN, and MLP, are trained to determine whether a truck is empty or a loaded platform semi-trailer or other body type (e.g., dry van, container, tank, open-top van/dump, auto transporter, etc.). Overall, the machine learning models produce relatively high accuracies; the result of the KNN model on test data identifies empty platform semi-trailers with 99% accuracy.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this research, methods are developed to classify truck-trailers and for detecting empty and loaded platform semi-trailers using side-fire LiDAR data. The presented methodology for processing the data involves a series of statistical models, heuristics, and machine learning algorithms to extract pertinent features to distinguish between different truck body types. Three different speed estimation methods are proposed using LiDAR data.

After extracting key features, various ML algorithms, including SVM, AdaBoost, KNN, and MLP, are trained to determine whether the truck in question is hauling a dry van, refrigerated van, 20 ft or 40 ft intermodal container, 40 ft refrigerated container, platform, tank, car transporter, open-top van/dump, or other. The other types of trailers contain various body types, including livestock, logging, custom-designed trailer, etc. While all ML methods produce relatively high accuracies, the results of the SVM model on test data are slightly higher with a medium CCR of 94%. In addition to the truck-trailer classification, empty and loaded platform semi-trailers are detected with 99% accuracy.

While installing multi-array 3D LiDAR sensors on highways is not a common practice today, it is anticipated that such sensors could become a viable option for data collection in the near future as they become more widely available and affordable. This study shows how additional truck classification information could be extracted if data from such sensors are available. Clearly, the scope of this paper is limited in terms of the variety of truck body types considered. The LiDAR data cannot be used to identify empty trucks with closed body types, e.g. a dry van, container, tank, etc. However, the author believes that this research will ultimately enhance freight and commodity modeling research by providing a detailed breakdown of truck

body types at observation stations where a LiDAR sensor is installed. The cost of the used hardware which includes Velodyne VLP-16 LiDAR, surveillance cameras, a laptop computer, and peripherals is around ten thousand dollars.

One can consider the price as a barrier for large-scale deployment; however, the price of the LiDAR sensors is dropping due to the recent development of automated vehicles. In addition to price, a LiDAR sensor has data quality issues in bad weather (e.g. rain, snow, etc.). Laser firings are affected by the rain or snow. Currently, researchers are developing models and methods to use LiDAR data in precipitation conditions. Another limitation of the LiDAR data is that it generates relatively large data. It requires a large amount of hard drive space if the LiDAR data need to be stored. For example, 15 minutes of Velodyne VLP-16 LiDAR data is around 1GB. Since the LiDAR data is relatively large, analyzing or processing requires more computational power.

For future work, new models could be developed to extract additional information from the cloud point data. For example, the third method for speed estimation could be extended to track individual axles. This will allow classifying vehicles based on the number of axles. In addition, new methods can be developed to mitigate the effects of precipitation conditions on LiDAR data. Finally, additional vehicle or truck classes can be considered to enrich the types of vehicles in the classification scheme.

REFERENCES

1. FHWA, *Traffic Monitoring Guide*. 2016.
2. Basar, G., M. Cetin, and A.P. Nichols, *Comparison of vehicle re-identification models for trucks based on axle spacing measurements*. Journal of Intelligent Transportation Systems, 2018. **22**(6): p. 517-529.
3. Cetin, M., C.M. Monsere, and A.P. Nichols, *Bayesian Models for Reidentification of Trucks Over Long Distances on the Basis of Axle Measurement Data*. Journal of Intelligent Transportation Systems, 2011. **15**(1): p. 1-12.
4. AASHTO *Freight Data Guide for Improved Transportation Planning*. 2018.
5. He, Z., et al., *Vehicle sensor data-based transportation research: Modeling, analysis, and management*. Journal of Intelligent Transportation Systems, 2019. **23**(2): p. 99-102.
6. Aijazi, A.K., et al. *Automatic detection of vehicles at road intersections using a compact 3D Velodyne sensor mounted on traffic signals*. in *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016.
7. Sazara, C., R.V. Nezafat, and M. Cetin. *Offline reconstruction of missing vehicle trajectory data from 3D LIDAR*. in *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017.
8. Sun, Y., et al., *3-D Data Processing to Extract Vehicle Trajectories from Roadside LiDAR Data*. Transportation Research Record, 2018. **0**(0): p. 0361198118775839.
9. Aycard, O., et al. *Intersection safety using lidar and stereo vision sensors*. in *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011.
10. Khattak, A., S. Hallmark, and R. Souleyrette, *Application of Light Detection and Ranging Technology to Highway Safety*. Transportation Research Record: Journal of the Transportation Research Board, 2003. **1836**: p. 7-15.
11. Mokhtarimousavi, S., et al., *Improved Support Vector Machine Models for Work Zone Crash Injury Severity Prediction and Analysis*, in *Transportation Research Board 98th Annual Meeting*. 2019: Washington DC, United States.
12. Parsa, A.B., et al., *Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis*. Accident Analysis & Prevention, 2020. **136**: p. 105405.
13. Ahmed, U., O. Sahin, and M. Cetin, *Minimizing GPS Dependency for a Vehicle's Trajectory Identification by Using Data from Smartphone Inertial Sensors and Onboard Diagnostics Device*. Transportation Research Record, 2017. **2644**(1): p. 55-63.
14. Ahmed, U., O. Sahin, and M. Cetin. *Minimizing GPS Dependency for a Vehicle's Trajectory Identification by Utilizing Data from Smartphone Inertial Sensors and OBD Device*. in *Transportation Research Board 96th Annual Meeting*. 2017. Washington DC.
15. Cetin, M., I. Ustun, and O. Sahin, *Classification Algorithms for Detecting Vehicle Stops from Smartphone Accelerometer Data*. Transportation Research Board, Washington DC, 2016.
16. Ustun, I., et al. *Detecting vehicle stops from smartphone accelerometer data*. in *The 21st World Congress on Intelligent Transportation Systems, ITS WC2014*. ITS WC. 2014.
17. Ustun, I. and M. Cetin, *Speed Estimation using Smartphone Accelerometer Data*. Transportation Research Record, 2019. **2673**(3): p. 65-73.

18. Paleti, R., O. Sahin, and M. Cetin, *Modeling the impact of latent driving patterns on traffic safety using mobile sensor data*. Accident Analysis & Prevention, 2017. **107**: p. 92-101.
19. Sahin, O., R. Paleti, and M. Cetin, *Investigating Relationship Between Driving Patterns and Traffic Safety Using Smartphones Based Mobile Sensor Data*, O.D. University, Editor. 2016.
20. Cetin, M., et al., *Feasibility of Estimating Commodity Flows on Highways with Existing and Emerging Technologies*. 2019.
21. Vatani Nezafat, R., S. Behrouz, and M. Cetin, *Classification of truck body types using a deep transfer learning approach*, in *The 21st IEEE International Conference on Intelligent Transportation Systems*. 2018.
22. Zhang, G., R. Avery, and Y. Wang, *Video-Based Vehicle Detection and Classification System for Real-Time Traffic Data Collection Using Uncalibrated Video Cameras*. Transportation Research Record: Journal of the Transportation Research Board, 2007. **1993**: p. 138-147.
23. Gupte, S., O. Masoud, and P. Papanikolopoulos. *Vision-based vehicle classification*. in *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493)*. 2000.
24. Chen, Z., T. Ellis, and S.A. Velastin. *Vehicle detection, tracking and classification in urban traffic*. in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. 2012.
25. Chung-Lin, H. and L. Wen-Chieh. *A vision-based vehicle identification system*. in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*. 2004.
26. Coifman, B., et al. *Roadway traffic monitoring from an unmanned aerial vehicle*. IEE Proceedings - Intelligent Transport Systems, 2006. **153**, 11-20.
27. Khan, S.M., et al. *3D model based vehicle classification in aerial imagery*. in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010.
28. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005.
29. Harlow, C. and S. Peng, *Automatic vehicle classification system with range sensors*. Transportation Research Part C: Emerging Technologies, 2001. **9**(4): p. 231-247.
30. Lee, H. and B. Coifman, *Side-Fire Lidar-Based Vehicle Classification*. Transportation Research Record: Journal of the Transportation Research Board, 2012. **2308**: p. 173-183.
31. Sandhawalia, H., et al. *Vehicle type classification from laser scanner profiles: A benchmark of feature descriptors*. in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. 2013.
32. Magnier, V., D. Gruyer, and J. Godelle. *Automotive LIDAR objects Detection and Classification Algorithm Using the Belief Theory*. in *IV 2017 - IEEE Intelligent Vehicles Symposium*. 2017. Los Angeles, United States: Institute of Electrical and Electronics Engineers - IEEE.
33. Sahin, O., R.V. Nezafat, and M. Cetin, *Classification of Truck Trailers Based on Side-Fire LIDAR Data*, in *Transportation Research Board 98th Annual Meeting*. 2019: Washington DC, United States.

34. Lee, H. and B. Coifman, *Using LIDAR to Validate the Performance of Vehicle Classification Stations*. Journal of Intelligent Transportation Systems, 2015. **19**(4): p. 355-369.
35. Xiao, W., et al., *Street-side vehicle detection, classification and change detection using mobile laser scanning data*. ISPRS Journal of Photogrammetry and Remote Sensing, 2016. **114**: p. 166-178.
36. 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, 2016. **68**: p. 1-21.
37. 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, 2019. **1**(1): p. 71-82.
38. Asborn, M.I., C.G. Burris, and S. Hernandez, *Truck Body-Type Classification using Single-Beam Lidar Sensors*. Transportation Research Record, 2019. **2673**(1): p. 26-40.
39. Holguín-Veras, J. and E. Thorson, *Modeling commercial vehicle empty trips with a first order trip chain model*. Transportation Research Part B: Methodological, 2003. **37**(2): p. 129-148.
40. Holguín-Veras, J., et al., *Commercial Vehicle Empty Trip Models With Variable Zero Order Empty Trip Probabilities*. 2010. **10**(2): p. 241-259.
41. Tatineni, V.C. and M.J. Demetsky, *Supply chain models for freight transportation planning*. 2005.
42. Mesa-Arango, R., S. Ukkusuri, and I. Sarmiento, *Network Flow Methodology for Estimating Empty Trips in Freight Transportation Models*. 2013. **2378**(1): p. 110-119.
43. Office, F.T.D.a.A., *Truck Empty Backhaul*. 2018.
44. FFMPEG, *FFMPEG Tool*. 2018.
45. Velodyne. *VLP-16 Puck*. 2018 [cited 2018; Available from: <http://velodynelidar.com/vlp-16.html>].
46. Hough, P.V., *Method and means for recognizing complex patterns*. 1962, Google Patents.
47. Duda, R.O. and P.E. Hart, *Use of the Hough transformation to detect lines and curves in pictures*. Communications of the ACM, 1972. **15**(1): p. 11-15.
48. Bishop, C.M. and N.M. Nasrabadi, *Pattern Recognition and Machine Learning*. J. Electronic Imaging, 2007. **16**.
49. Dasarathy, B.V., *Nearest neighbor (NN) norms : NN pattern classification techniques*. IEEE Computer Society Tutorial, 1991.
50. Bishop, C.M., *Neural networks for pattern recognition*. 1995: Oxford university press.
51. Eibl, G. and K.-P. Pfeiffer, *Multiclass boosting for weak classifiers*. Journal of Machine Learning Research, 2005. **6**(Feb): p. 189-210.
52. Burges, C.J., *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery, 1998. **2**(2): p. 121-167.
53. R Core Team, *R: A Language and Environment for Statistical Computing*. 2010, R Foundation for Statistical Computing: Vienna, Austria.
54. Kuhn, M., *Building Predictive Models in R Using the caret Package*. Journal of Statistical Software, Articles, 2008. **28**(5): p. 1--26.

APPENDIX A

DETAILED BREAK-DOWN OF LABELLED TRUCK-TRAILERS

Category	Hernandez, Tok [36] Body Class	Labelled Sample Size
Van	Enclosed van	2,591
	Skirted enclosed van	54
	Drop frame van	18
	Reefer enclosed van	1,552
Tank	Hot product tank	6
	Deep drop tank	
	Food grade tank	23
	Petroleum tank	180
	Chemical tank	4
	Crude oil tank	6
	Air compression tank	9
	Propane tank	3
	Pneumatic Tank	160
Platform	Basic	949
	Step deck	379
	Low boy	200
Specialty	Bottom/Belly dump	
	Bulk waste transport	
	End dump	232
	Livestock	1
	Curtainside van	
	Open top van	14
	Pole, logging, pulpwood, or pipe	7
	Automobile transport	378
	Beverage	
	Hopper	
	Agricultural van	
Intermodal Containers	Container chassis	45
	40 ft container	1,919
	40 ft refrigerated container	251
	45 ft container	46
	20 ft container	212+91
	20 ft container on 40 ft chassis	42
	53 ft container	32
Small Trailers	Recreational vehicle trailer	124
	Towed vehicle	63

APPENDIX B

TRUCK-TRAILER IMAGES AND CONSTRUCTED 3D PROFILES FROM LiDAR DATA

In Appendix A, available truck-trailers from collected data will be shown. To select a truck-trailer from collected data, HPMS based classification method has been used. When a grouping method needed to apply, HPMS recommends a maximum of six generalized vehicle classes.

These are:

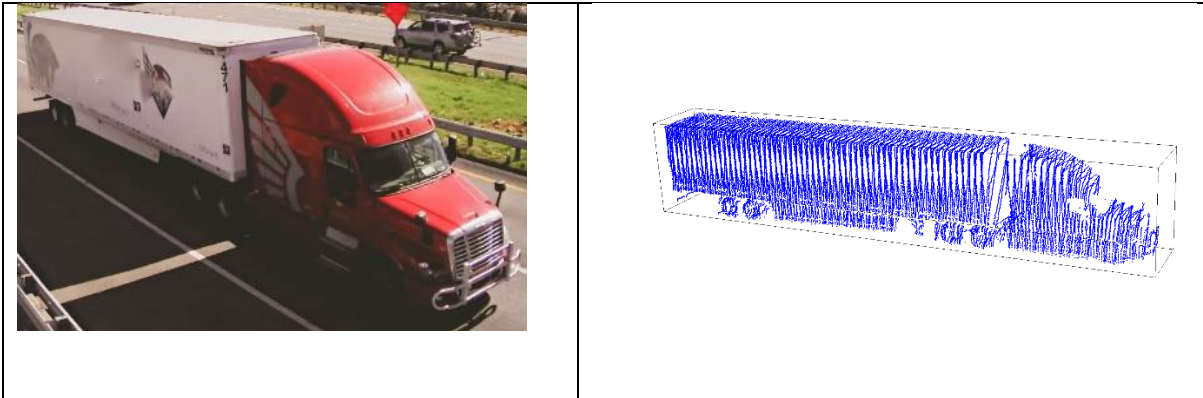
1. Motorcycle (MC)
2. Passenger vehicles under 102” (PV)
3. Light trucks over 102” (LT)
4. Buses (BS)
5. Single unit trucks (SU)
6. Combination trucks (tractor-trailers) (CU)

Based on this recommendation and nature of this study, group six vehicles are selected from the data. Then vehicle configuration of these vehicles and trailers updated using developed UI.

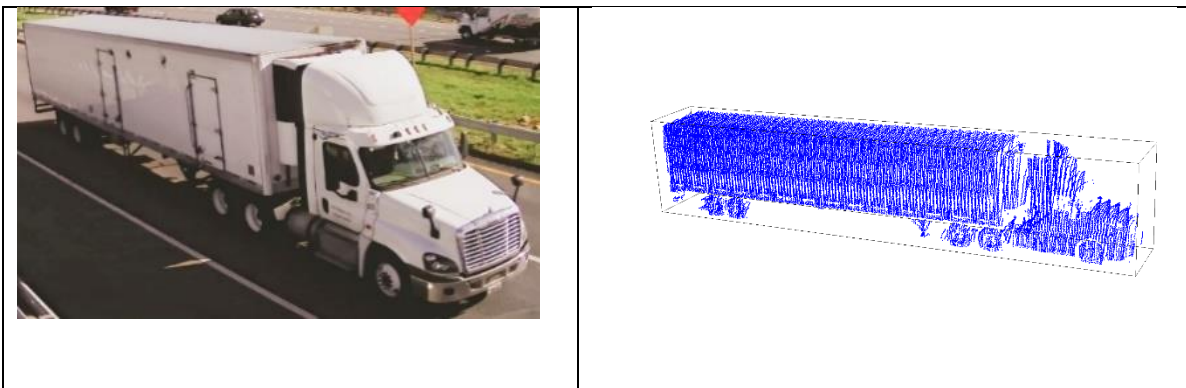
In the trucking industry, truck-trailers are also grouped within their type. For example, intermodal containers contain different lengths such that; 20, 40, 45, and 53 ft; flat-bed trailers are basic, lowboy, step deck, then there is a different type of tankers and dry vans. Truck-trailer images with their constructed 3D profiles are shown in this appendix.

Enclosed Vans

53 ft Dry Van

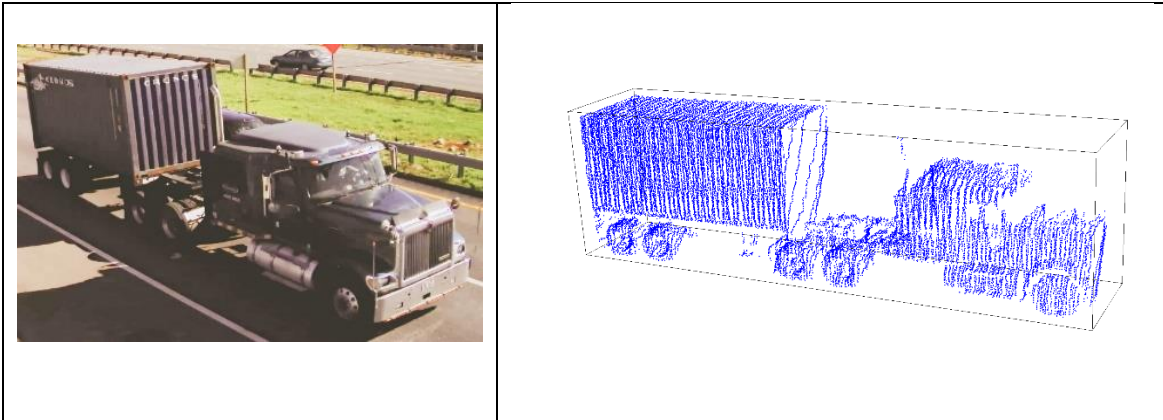


53 ft Refrigerated Dry Van



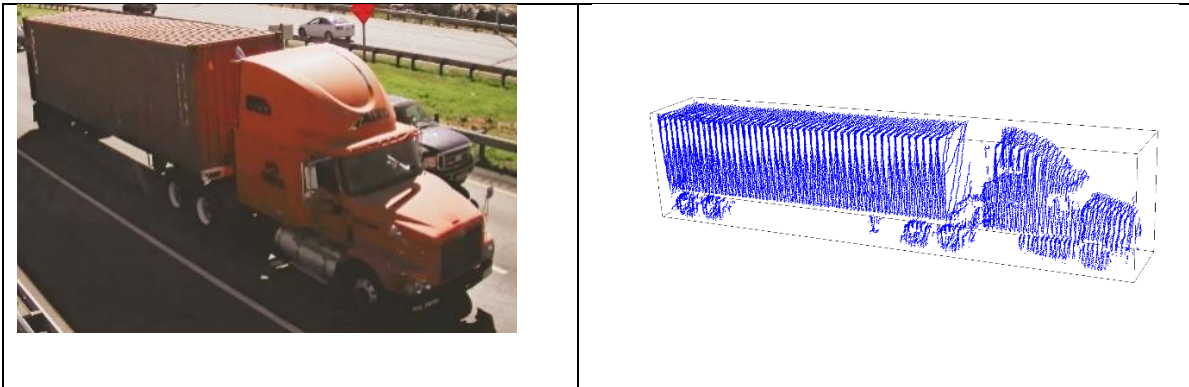
Intermodal Containers

20 ft Container

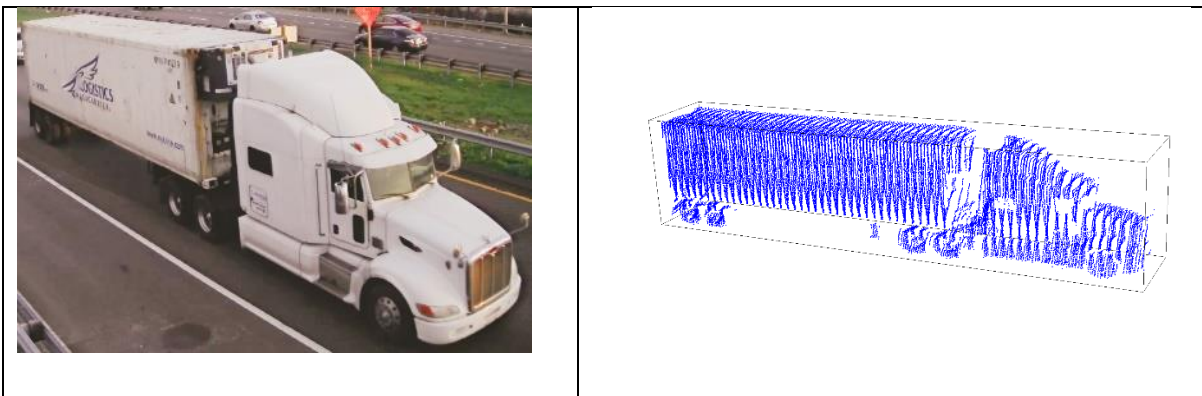


20 ft Reefer Container

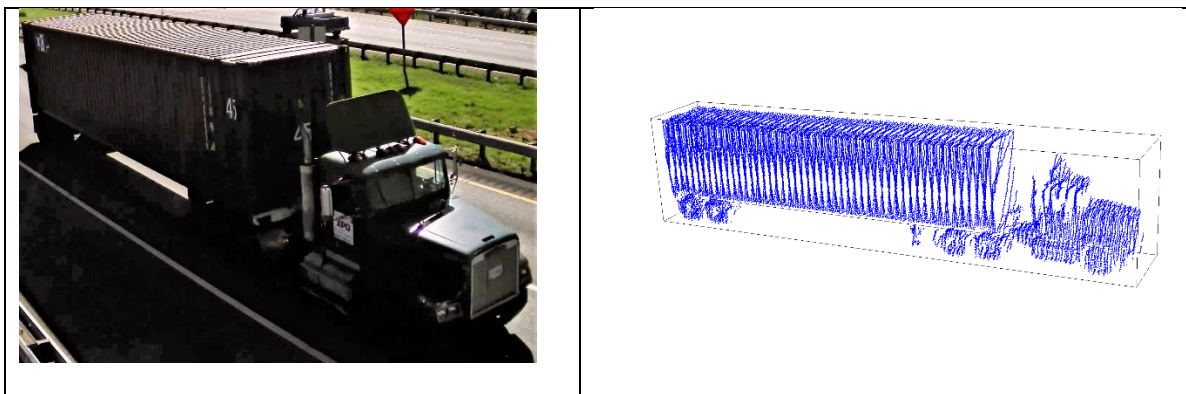
40 ft Container



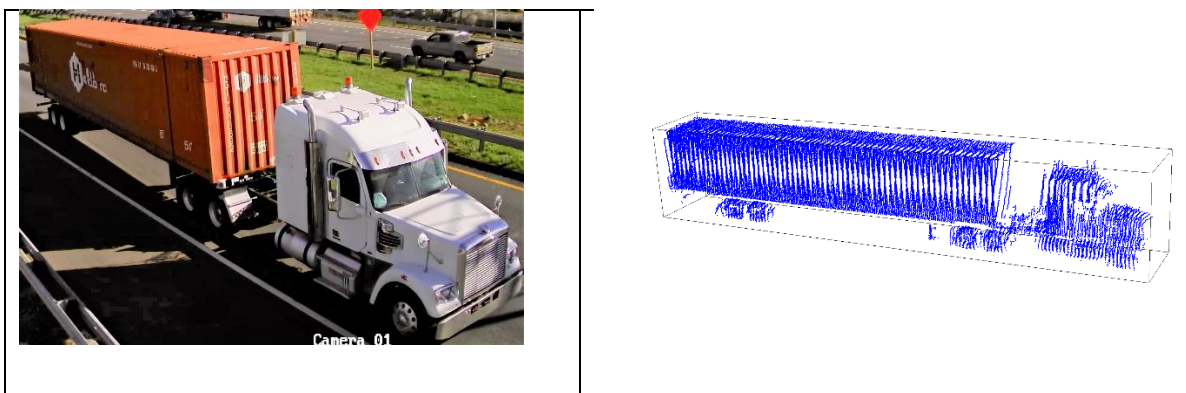
40 ft Reefer Container



45 ft Container

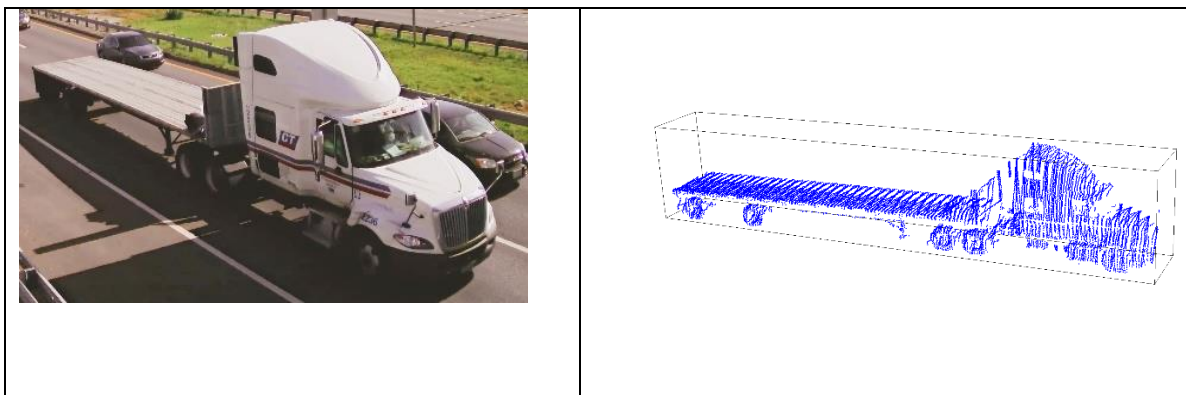


53 ft Container

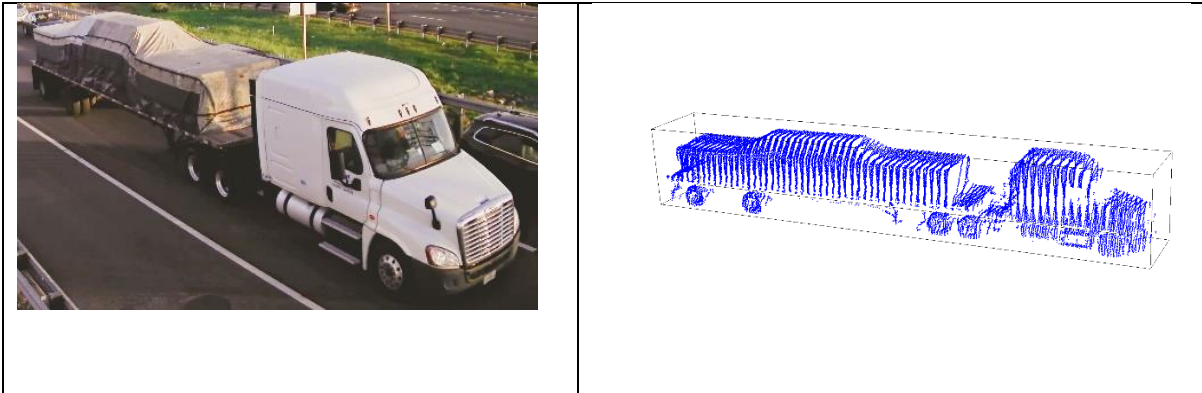


Platforms

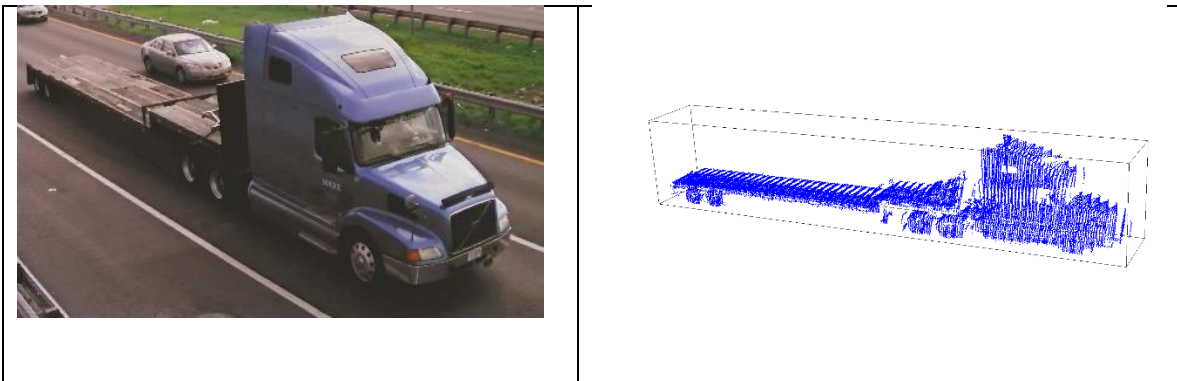
Empty Basic Platform



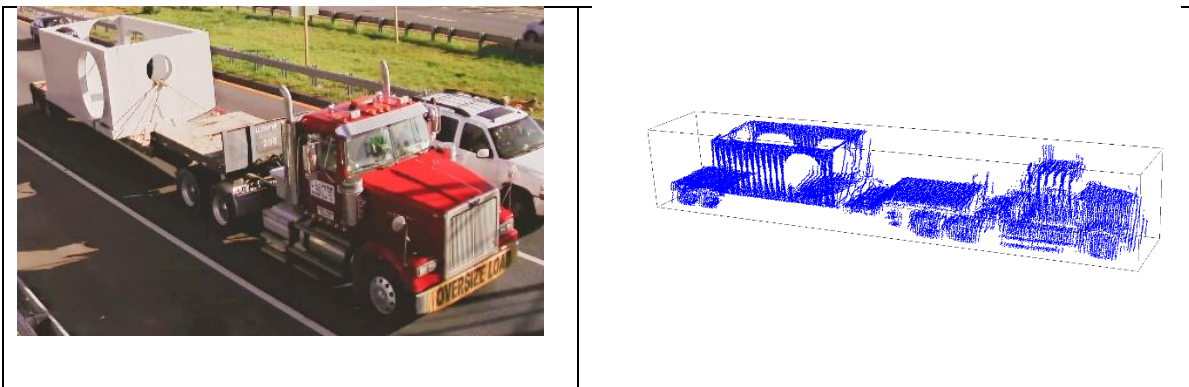
Loaded Basic Platform



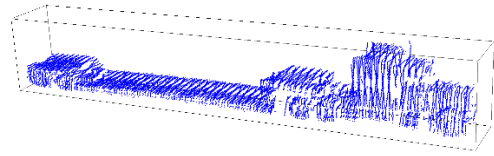
Empty Step Deck Platform



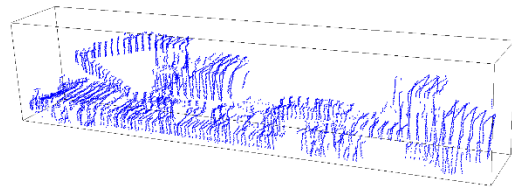
Loaded Step Deck Platform



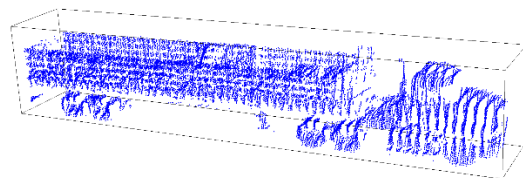
Empty Low Boy Platform



Loaded Low Boy Platform

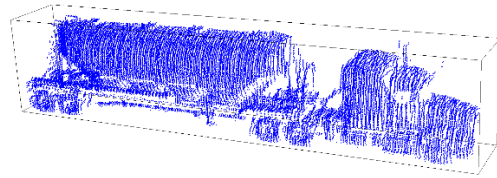


Stake Body Platform

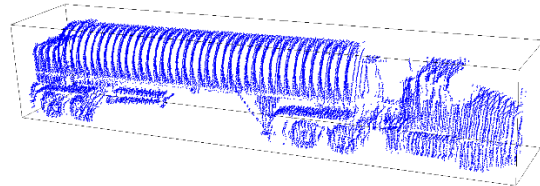


Tanks

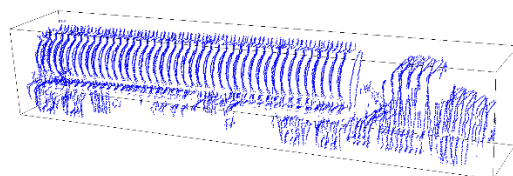
Pneumatic Tank



Cylinder Tank

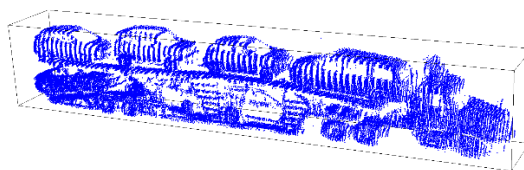


Ellipse Tank

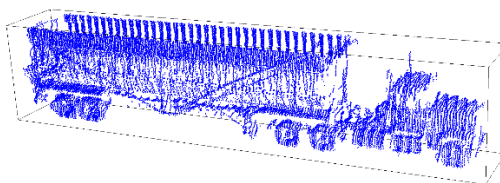


Specialty

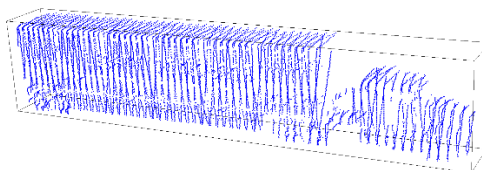
Automobile Transport



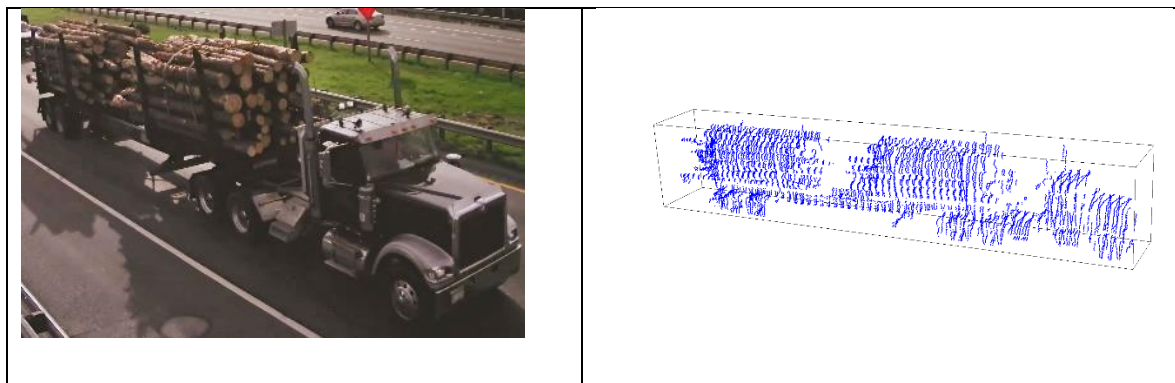
Open Top/Dump (End)



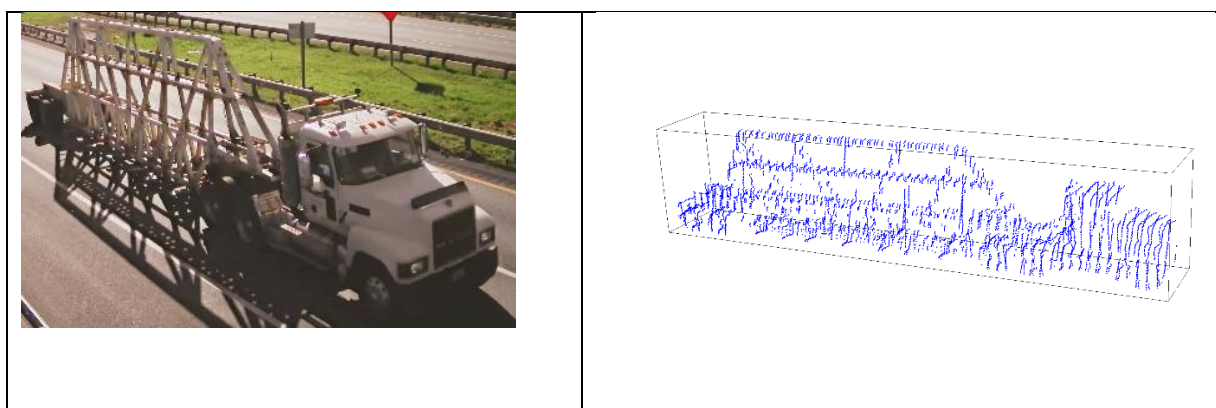
Livestock



Logging



Custom Trailer



VITA

Olcay Sahin
Department of Civil and Environmental Engineering
Old Dominion University
Norfolk, VA 23529

Olcay Sahin received his bachelor's degree in Industrial Engineering from International Black Sea University in Tbilisi Georgia in 2005, a Master of Business Administration degree from Strayer University in Anne Arundel, Maryland in 2008, and a Master of Science degree from the Department of Computational Modeling and Simulation Engineering at Old Dominion University in Norfolk, Virginia in 2012. He has served as a graduate research assistant for the Transportation Research Institute (TRI) at Old Dominion University in Norfolk, Virginia. His research interests include transportation systems modeling, agent-based simulation, artificial intelligence, machine learning, and intelligent transportation systems (ITS).