Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

DEVELOPMENT AND EVALUATION OF LOW COST 2-D LiDAR BASED TRAFFIC DATA COLLECTION METHODS

by
Ravi Jagirdar

Traffic data collection is one of the essential components of a transportation planning exercise. Granular traffic data such as volume count, vehicle classification, speed measurement, and occupancy, allows managing transportation systems more effectively. For effective traffic operation and management, authorities require deploying many sensors across the network. Moreover, the ascending efforts to achieve smart transportation aspects put immense pressure on planning authorities to deploy more sensors to cover an extensive network. This research focuses on the development and evaluation of inexpensive data collection methodology by using two-dimensional (2-D) Light Detection and Ranging (LiDAR) technology. LiDAR is adopted since it is economical and easily accessible technology. Moreover, its 360-degree visibility and accurate distance information make it more reliable.

To collect traffic count data, the proposed method integrates a Continuous Wavelet Transform (CWT), and Support Vector Machine (SVM) into a single framework. Proof-of-Concept (POC) test is conducted in three different places in Newark, New Jersey to examine the performance of the proposed method. The POC test results demonstrate that the proposed method achieves acceptable performances, resulting in 83% ~ 94% accuracy. It is discovered that the proposed method's accuracy is affected by the color of the exterior surface of a vehicle since

some colored surfaces do not produce enough reflective rays. It is noticed that the blue and black colors are less reflective, while white-colored surfaces produce high reflective rays.

A methodology is proposed that comprises K-means clustering, inverse sensor model, and Kalman filter to obtain trajectories of the vehicles at the intersections. The primary purpose of vehicle detection and tracking is to obtain the turning movement counts at an intersection. A K-means clustering is an unsupervised machine learning technique that clusters the data into different groups by analyzing the smallest mean of a data point from the centroid. The ultimate objective of applying K-mean clustering is to identify the difference between pedestrians and vehicles. An inverse sensor model is a state model of occupancy grid mapping that localizes the detected vehicles on the grid map. A constant velocity model based Kalman filter is defined to track the trajectory of the vehicles. The data are collected from two intersections located in Newark, New Jersey, to study the accuracy of the proposed method. The results show that the proposed method has an average accuracy of 83.75%. Furthermore, the obtained R-squared value for localization of the vehicles on the grid map is ranging between 0.87 to 0.89.

Furthermore, a primary cost comparison is made to study the cost efficiency of the developed methodology. The cost comparison shows that the proposed methodology based on 2-D LiDAR technology can achieve acceptable accuracy at a low price and be considered a smart city concept to conduct extensive scale data collection.

DEVELOPMENT AND EVALUATION OF LOW COST 2-D LiDAR BASED TRAFFIC DATA COLLECTION METHODS

by
Ravi Jagirdar

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Transportation

John A. Reif, JR. Department of Civil and Environmental Engineering

AUGUST 2020

APPROVAL PAGE

DEVELOPMENT AND EVALUATION OF LOW-COST TRAFFIC DATA COLLECTION METHODS

Ravi Jagirdar

Dr. Joyoung Lee, Dissertation Advisor — Date
Associate Professor of Civil and Environmental Engineering, NJIT

Dr. Steven I-Jy Chien, Committee Member — Date
Professor of Civil and Environmental Engineering, NJIT

Dr. Janice R. Daniel, Committee Member — Date
Professor of Civil and Environmental Engineering, NJIT

Dr. Branislav Dimitrijevic, Committee Member — Date
Assistant Professor of Civil and Environmental Engineering, NJIT

Dr. Jing Jin, Committee Member — Date
Assistant Professor of Civil and Environmental Engineering, Rutgers - New Brunswick

BIOGRAPHICAL SKETCH

Author: Ravi Jagirdar

Degree: Doctor of Philosophy

Date: August 2020

Undergraduate and Graduate Education:

- Doctor of Philosophy in Transportation,
  New Jersey Institute of Technology, Newark, NJ, 2020

- Master of Science in Transportation,
  New Jersey Institute of Technology, Newark, NJ, 2015

- Bachelor of Engineering in Civil Engineering,
  Gujarat Technological University, Gujarat, India, 2013

Major: Transportation

Publications and Presentations:

R. Jagirdar, J. Lee, K. Kim, M. Kang, "Development and Evaluation of Traffic Count Sensor Using LiDAR and Continuous Wavelet Transform" *in 98th Transportation Research Board Annual Meeting* 2019, Washington, DC

R. Jagirdar, J. Lee, K. Kim, "Development of Low-Cost Traffic Sensors Using 2-Dimensional LiDAR for Smart City" in *American Society of Civil Engineering International Conference on Transportation and Development* 2018, Pittsburgh, Pennsylvania

R. Jagirdar, J. Lee, "Development and Evaluation of Real-Time Online Simulation Framework" in *Intelligent Transportation Systems World Congress* 2017, Montreal, Canada

J. Lee, R. Jagirdar, "Development of On-Line Simulation Framework for Proactive Traffic Management Strategies" in *17th Annual North America PTV Vision User Group meeting* 2016, Arlington, Virginia

J. Lee, R. Jagirdar, "Development of On-Line Simulation Framework for Proactive Traffic Management Strategies" in *Intelligent Transportation Society of New Jersey Annual Meeting* 2016, New Brunswick, NJ

J. Lee, R. Jagirdar, “Work Zone Congestion Estimation Using an Online Simulation Framework” in *18th New Jersey Department of Transportation (NJDOT) Research Showcase* 2016, West Windsor, NJ

K. Kim, J. Lee, R. Jagirdar, “Statewide Work Zone Monitoring Systems” in *17th New Jersey Department of Transportation (NJDOT) Research Showcase* 2015, West Windsor, NJ

This dissertation is dedicated to my family:

Brother, Mihir Jagirdar

Sister-in-law, Hemali Bhrahmbhatt, and

Parents, Hemant and Mandakini Jagirdar

For all their support

ACKNOWLEDGMENT

I want to express my most profound appreciation to all who helped me and provided their advice to complete my doctoral study.

First, I would like to thank my advisor Dr. Joyoung Lee for his guidance, encouragement, and advice throughout my doctoral study. His perseverance and problem-solving skills always inspired me to deal with the problems I faced during research. Without his support, this dissertation would not have been possible. I would also like to thank the New Jersey Department of Transportation Intelligent Transportation System (ITS) Resource Center at NJIT for financial support and allowing me to participate in several projects. The skills and experience I gained through those projects were of irreplaceable importance for me. I want to thank Dr. Dejan Besenski, Dr. Taha Marhaba, and Dr. Steven Chien for their guidance and support related to administrative work at NJIT.

I would also like to acknowledge my sincere appreciation to the rest of my dissertation committee, Dr. Steven Chien in the Department of Civil and Environmental Engineering at NJIT, Dr. Jing (Peter) Jin of the Rutgers School of Engineering - New Brunswick, and Dr. Janice R Daniel in the Department of Civil and Environmental Engineering at NJIT valuable input.

Many thanks to Dr. Zijia (Gary) Zhong, Dr. Chaitanya (CP) Pathak, and Dr. Slobodan Gutesa for constructive discussions and knowledge sharing.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF ACRONYMS

USDOT	United Department of Transportation
TCM	Transportation Congestion Management
FHWA	Federal Highway Administration
IoT	Internet of Things
ITS	Intelligent Transportation System
RTMS	Remote Traffic Microwave Sensors
VIP	Video Image Processing
GPS	Global Positioning System
2-D	2 – Dimensional
LiDAR	Light Detection and Ranging
3-D	3 – Dimensional
CWT	Continuous Wavelet Transform
SP model	Symmetries and Perturbation Model
EKF	Extended Kalman Filter
DGPS	Differential Global Positioning System
CAV	Connected and Autonomous Vehicles
LLD	LiDAR Lane Detection
OLR	Optical Lane Recognition System
ACC	Adaptive Cruise Control
UKF	Unscented Kalman Filter
RANSAC	Random Sample Consensus
MC-DBSCAN	Multi Classified Density Based Spatial Clustering Method
INS	Inertial Navigation System
DoN	Difference of Normals
IMU	Inertial Measurement Unit
TScan	Traffic Scanner
DWT	Discrete Wavelet Transform
DDWT	Discrete Dyadic Wavelet Transform
ACF	Autocorrelation Function
DWPT	Discrete Wavelet Packet Transform

SLAM	Simultaneous Localization and Mapping
GNN	Global Nearest Neighbor
JPDA	Joint Probabilistic Data Association
LSTM	Long Short-Term Memory
UAV	Unmanned Aerial Vehicle
CV	Computer Vision
IMM	Internal Multiple Model
GIS	Geographic Information System
ATM	Advanced Triangulation Method
RMSE	Root Mean Squared Error
AV	Autonomous Vehicle
CNN	Convolution Neural Network
GMM	Gaussian Mixture Model
RGB	Red-Green-Blue
FOV	Field of View
POC	Proof of Concept
NB	North Bound
SB	South Bound
WB	West Bound
EB	East Bound

CHAPTER 1

INTRODUCTION

1.1 Background

By 2025, nearly 70% of the world population will settle down in urban areas [1, 2]. The latest U.S. Census Bureau data showed that most of the largest cities in the U.S. experienced population growth between 2015 and 2016 [3]. The population expansion put immense stress on the cities since it demands sophisticated transportation facilities, a healthy economy, a stable water/ power supply system, and a quality environment. “Smart City” is one of the feasible options for city planning commissions to fulfill growing requirements. Over the last decade, many regional planning authorities intensify their efforts to achieve the “Smart City” title. A smart city is defined as an urban area that uses various types of electronic data collection sensors to collect vital information, which is used to manage assets and resources efficiently. The smart city includes data collection from multiple means to monitor and manage traffic and transportation systems, power plants, water supply networks, and waste management. According to the experts, over the next 20 years of the period, cities around the world will invest approximately $41 trillion to upgrade their infrastructure [4, 5]. ‘Smart City’ technology enables to improve everything from the quality of the air and water to transportation, energy, and communication systems by integrating cutting-edge technology and resources.

Transportation is one of the vital features of cities’ growth and success. In 2015, the United States Department of Transportation (USDOT) announced a smart city challenge to support advanced technology implication in transportation

projects [4]. The primary purpose of the initiative is to endorse the advanced technology applications in Transportation and Congestion Management (TCM) program run by the Federal Highway Administration (FHWA). An efficient traffic data collection is an essential requirement for effective transportation system management and operations. Traffic data study helps to improve the efficiency of the road network and to provide better means for the development of infrastructures. Traffic data such as volume, vehicle classification, speed, axle weights, and gross vehicle weights provide definitive information for transportation planners and pavement design engineers.

The application of sensors and data visualization is one of the fundamental exercises carried out by regional planning authorities to collect diversified traffic data. In the process of achieving Smart City / Smart Transportation aspects, it is essential to obtain real-time information through deployed sensors and analyze it through open data portals that allow controlling cities' various operations pro-actively. Commonly, smart cities use the Internet of Things (IoT) devices such as connected sensors, which can collect real-time data related to lights, atmosphere, air quality, traffic, and analyze it. Business Insider Intelligence predicts that nearly 24 billion IoT devices will be installed across the world by 2020, and the investment in IoT devices will be $6 trillion over the next four and a half years [6]. Market reports estimate an annual growth rate of 25% in smart transportation for the coming five years [6].

With the drastic advancement of Intelligent Transportation Systems (ITS) and communications technologies, the data collection techniques improve very

rapidly over the last decade. Based on the installation of the detectors/sensors, the sensors can be classified into three categories, such as 1) intrusive, 2) non-intrusive, and 3) off-roadway. The intrusive methods consist of data recorder and sensors placed on or in the pavement surface (e.g., pneumatic tube counter, piezoelectric sensors, and magnetic loop). The non-intrusive methods consist of the application of radar technology, Remote Traffic Microwave Sensor (RTMS), Video Image Processor (VIP), etc., which are often very expensive. Thus, the higher cost of installation and maintenance would not allow a large number of sensor deployment. In the off-roadway method, the data is collected from probe vehicles, which comprises of the advanced global positioning system (GPS), mobile phones, and Bluetooth/Wi-Fi-enabled devices. While the probe vehicle-based technique is more cost-effective compared to intrusive and non-intrusive detectors, it is also incapable of directly estimating traffic flow (e.g., counts) on a designated roadway segment.

1.2 Problem Statement

Smart transportation is one of the critical features of the smart city, which manages vehicular traffic to allow people and goods to be moved quickly through various means. ITS is one of the examples of smart transportation that deploy numerous amounts of sensors to collect data. The current technology to collect traffic data consists of RTMS, Video Analytics, loop detectors, and pneumatic tube counters. With the current technology, it would be expensive and challenging to deploy sensors on a large scale. Thus, it is necessary to have a traffic data collection technology that allows freeway and arterial traffic management centers to equip

the road networks at ease and at a low cost. Table 1.1 shows the current technologies and their pros and cons.

Table 1.1 Current Traffic Data Collection Technologies

	Installation problems		Type of data collection	
Technology	Lane closure	Pavement work	Traffic count	Vehicle trajectory
Pneumatic Tube	√	×	√	×
RTMS	×	×	√	×
Video Analytics	×	×	√	√
Loop Detector	√	√	√	×
Piezoelectric Sensors	√	√	√	×

The main goal of the dissertation is to develop methodologies that can handle two – dimensional (2-D) Light Detection and Ranging Technology (LiDAR) data to conduct traffic count and vehicle trajectory extraction. Many studies have been conducted to employ LiDAR or RADAR technology to collect traffic data such as vehicle count, vehicle classification, and vehicle trajectories at the signalized intersection. From the conducted literature review, it is found that the previous research studies used expensive LiDAR technology, with vision sensors (cameras). Furthermore, the LiDAR's used in the prior studies deals with three-dimension (3-D) point clouds with high sampling rates that ease the data processing burden.

Currently, LiDAR technology is gaining more attention in the autonomous car and drone industries. Increased application of the ground-based and airborne LiDAR pushes manufacturers to introduce low-cost LiDAR to the market. Although the low-cost LiDAR cannot scan the surrounding environment in detail due to a low level of emitter/receiver pairs or "channels." A higher number of channel LiDAR provides detailed information about the surrounding environment since it can produce tens or hundreds of thousands of lasers pulses every second. The primary focus of the developed methodology is to achieve the high accuracy of data using low-cost LiDAR and collect a useful traffic data set.

1.3 Goal and Objectives

The primary goal of this dissertation is to Develop cost-effective traffic data collection methods using 2–D LiDAR sensors that allow city authorities to conduct traffic data collection on a large scale at a low cost and can contribute to achieving the "Smart Cities" concept. To this end, the following objectives are established:

- To develop a methodology to acquire traffic count by analyzing 2-D LiDAR data.
- To develop a methodology to construct vehicle trajectories for turning moment count at the signalized intersections using 2–D LiDAR.
- Investigate the applicability and effectiveness of proposed methodology using the real-world data.

The rest of this dissertation is organized as follows: Chapter 2 covers the comprehensive literature review with respect to 1) LiDAR application in transportation engineering for data collection, 2) Application of CWT for signal analysis, 3) Inverse sensor model, and 4) Application of Kalman filter to track the

objects. Chapter 3 discusses the various components of the proposed methodologies in-depth, followed by framework development in Chapter 4. The conducted proof-of-concept tests are explained in Chapter 5. Chapter 6 discusses the results in detail obtained from the real-world data, and Chapter 7 provides conclusion and discussion with future research study.

CHAPTER 2
LITERATURE REVIEW

In this section, extensive literature reviews are conducted. The literature reviews are divided into four sections; Section 2.1 explains the previous research and commercial applications of LiDAR in transportation engineering to enhance the autonomous vehicle performance. Section 2.2 provides comprehensive literature review of application of LiDAR for data collection and tracking. Section 2.3 consists of a literature review of Continuous Wavelet Transform (CWT) application in the field of transportation and structural engineering. A CWT literature review provides an idea about CWT's appropriateness to capture the significant changes in the signal. In this dissertation, the primary purpose of using CWT is to deal with noisy LiDAR data and capture the specific distortion in the signal caused due to the presence of the vehicle. Section 2.4 comprises of inverse sensor model application with occupancy grid mapping in the field of robotics and autonomous vehicles. An inverse sensor model is applied in this study to localize detected objects. Section 2.5 includes the literature review of Kalman Filter's application to track an object, which shows the efficiency of the Kalman Filter to obtain a trajectory from various sources of data. The collected data from the real world consists of noisy data, which makes the tracking task difficult. The Kalman filter's competence to predict correct data with the new measurement allows for tracking an object with noisy data and used in this dissertation.

2.1 LiDAR Applications in Robotics and Autonomous Vehicles

Over many years, various practical applications of LiDAR in different areas such as agriculture, aerial mapping, and 3-D structural mapping have been studied. Furthermore, over the last decade, much research has been done to utilize LiDAR information to improve autonomous vehicle performance by mapping the surrounding area in real-time.

Neira et al. [7] focus on the localization of a mobile robot in an indoor environment using information obtained from a range sensor. The primary objective of the study is to propose a methodology to fuse multiple range sensors for indoor localization of robot with symmetries and perturbation model (SP model), a probabilistic representation model, and an Extended Kalman Filter (EKF). The proposed method uses range points and intensity images obtained from the range scanner to locate the obstacle and identify the vertical edges of the obstacles. For the mobile robot trajectory, a prediction of the robot location is calculated by using previous odometric sensor reading fuse with current reading. The results show that the use of range and intensity information could be an impactful and robust approach for precise mobile robot localization.

Sergi et al. [8] investigate the different applications of LiDAR to improve the driver's visibility under hazardous conditions. The proposed method focuses on the development of a real-time computer-generated display that accurately displays the detected vehicle in the surrounding area. The system consists of a Differential Global Positioning System (DGPS) and LiDAR to locate the nearby vehicle precisely. The algorithm is divided into four different steps: 1) Data filtering

using threshold values, 2) Data clusters, 3) Vehicle Detection, and 4) Tracking. In the data filtering step, the unnecessary data collected from LiDAR is cleaned based on threshold values. Furthermore, the DGPS position is used to identify nearby road objects such as guardrails and mailboxes. The threshold value is calculated based on the closest distance obtained from LiDAR. The refined data is clustered into different clusters based on the distance to nearby points. The clustered data points are then analyzed to define the object as a vehicle or not based on the number of points in each cluster and the cluster's location in the LiDAR's field of view. Successful vehicle detection is defined by observing three different sections of the field of views of LiDAR, which is defined as the left, central, and right section, as shown in Figure 2.1. The clusters are compared to a set of criteria that is determined by the section in which the cluster resides. Vehicle tracking is done by checking each scanned cycle data, and a simple extrapolated technique is applied to calculate the movement of a detected vehicle at each scan. A table of tracked vehicles is created to track objects from scan to scan. If one of the current vehicle positions is within a certain tolerance of the position of an extrapolated tracked vehicle, then the data in the table is updated using the current vehicle information, and the speed and heading are recalculated based upon the current and previous data. The experiment results show that the LiDAR-based detection and tracking algorithm can accurately detect and track vehicles.

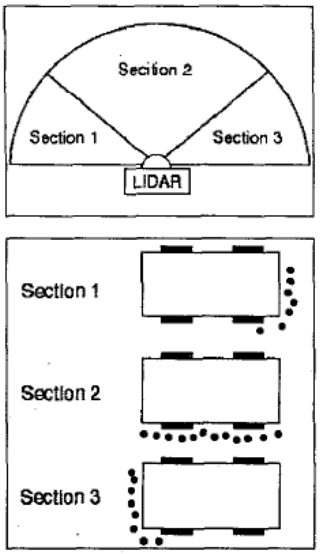


Figure 2.1 Defined section of field of view of LiDAR.
Source: [8]

Mendes et al. [9] develop the application of Laser range finder concerning autonomous vehicles to develop an effective collision avoidance system algorithm. The algorithm consists of three different steps: 1) Scan segmentation, 2) object classification, and 3) Object tracking using a Kalman filter. The scan segmentation consists of clustering and line fitting method to accurately define the generated cloud points are obtained from different objects. The object classification is based on the voting scheme that considers every hypothesis over time until it gets high classification confidence. The tracking of an object is done using a Kalman Filter by assuming constant velocity and acceleration. For each detected object, the Kalman filter is independently applied to every hypothesis of an object type, while the related object has not been classified in a specific class. The collision time is calculated using estimated racking results for each object. The system proved to

be efficient in tracking multi-objects over time, resulting in reasonable velocity estimates.

Reyher et al. [10] present an approach to use LiDAR for object and lane detection to enhance the lateral control of Connected and Autonomous Vehicles (CAV) named LiDAR Lane Detection (LLD). The proposed method consists of an EKF. The effectiveness of the proposed method is studied using real-world data. Data are collected by mounting two sensors on the front side of a vehicle. The results are compared with the Optical Lane Recognition system (OLR). The comparison shows that the results obtained from LLD are similar to OLR for straight and curved roads. The results summarize that the proposed approach can achieve the same accuracy as video referenced lane detection.

In 2006 Ogawa and Takagi [11] propose an approach that uses 2-D LiDAR data for lane recognition. The lane recognition is done using lane curvature, yaw angle, and offset data with Hough transformation. A Hough transformation is a computer vision-based approach to extract features from the images. An EKF algorithm is used to track the lane marks along the road surface. A DENSO's LiDAR is used, which emits the laser beam with six layers in the vertical direction and 451 scannings in a horizontal direction, which is mainly designed for Adaptive Cruise Control (ACC) application. The data is collected from the highway, and the results show the acceptable performance compared to the position of the lane marks in the forward image.

Kammel and Pitzer [12] develop a method for robust estimation to detect lane marker detection and mapping using multichannel 2-D LiDARs with stereo

cameras. The data obtained from LiDAR and vision cameras are integrated with global positioning navigation and odometry system that precisely provides location information of the vehicle. The collected LiDAR and stereo camera data are combined to create a 3-D map of the surrounding environment, and a multi-hypothesis approach is used to detect, cluster, and track a moving object. A lane marking is identified using LiDAR and camera images. Subsequent scans are registered and processed with GPS information to improve the detection of lane markings. The results suggest that the multichannel LiDAR and stereo camera information allows identifying lane robustly in the presence of shadows, against direct sunlight and even at night.

Linder et al. [13] propose multichannel LiDAR for lane marking extraction using EKF. The developed approach combines the intensity and distance information obtained from LiDAR. A polar grid cell is arranged circularly to develop a 3-D view of the detected data. Lane mark measurements are extracted using the lane detector grid, which is the combination of reflectivity and distance information. An unscented Kalman filter (UKF) is used to estimate the lane parameters, that utilized the raw lidar data. The difference between the road surface and lane markings are identified using an adaptive threshold technique. The proposed methodology is tested under various conditions. The results show that the road surface made from black asphalt is not reflective of the LiDAR signal as compared to the bright concrete surface. However, considering the lane marking design, it is found out that the developed methodology performed well with black road surface with raised lane markers.

To amplify the application of the LiDAR in a connected and autonomous vehicle, Shin et al. [14] propose a practical approach for lane marking detection using image and Velodyne HDL-32E LiDAR data. The algorithm works in three different steps; the first step extracts the ground information using the Random Sample Consensus (RANSAC) algorithm. A RANSAC is a repeated process of fitting a hypothesized plane to a set of 3-D data points and accepting points as inliers if their distances to the plane are below a threshold. The second steps identify the lane separately using 3-D LiDAR cloud points and image sensors. The lanes are defined from the LiDAR data by defining the intensity of points at each scanned azimuth, which is defined using experimental data sets. Intensity-based imagery sensors are used to identify the lane from the background information since the lane markings have a higher intensity than the pavement. The data processed from 3-D LiDAR and imagery sensors are combined in the third step to exclude false-positive points by observing the cross-relation between lanes identified using LiDAR and imagery sensor. The accurately detected lanes form the image coordinate are then projected to the LiDAR coordinate. The proposed approach is applied for straight roads, curves, intersection, and crosswalks. The results show that the integration of LiDAR and imagery sensor reduces the false positive detection and improved the accuracy.

Wu et al. [15] develop a methodology to apply roadside LiDAR to identify lanes and extract background information. The algorithm consists of a multi-classified density-based spatial clustering method (MC-DBSCAN). This research's primary objective is to provide a path to utilize roadside LiDAR information under

mixed traffic conditions. Here, the mixed traffic condition is described as a combination of connected and non-connected vehicles. A Velodyne VLP-16 LiDAR is used for this study. The LiDAR can generate 3-D cloud points for a 360-degree field of view, which can generate 600,000 3-D points per second. The collected LiDAR data is converted to cartesian coordinates (x,y,z) from spherical coordinates (r,ω,α) using know LiDAR position from GPS. The collected results show that the proposed method can filter the background, eliminate pedestrian movement around the intersection, and identify the lane position of vehicles.

2.2 LiDAR Applications for Data Collection and Tracking

Grejner-Brzezinska et al. [16] examine the feasibility of airborne LiDAR to collect traffic flow data. The study objectives are to develop and test the technique to identify and extract vehicles from LiDAR data and to develop methods for automatic vehicle classification. The algorithm extracts the vehicle information from LiDAR data by removing road surface data. The LiDAR data is collected along with GPS/ Inertial Navigation System (INS) sensor. Figure 2.2 shows the complete data processing architecture of the method. Estimated results show that the velocities of the large vehicles are estimated accurately, while the estimation of the small vehicle which is moving in the opposite direction of the LiDAR is critical.

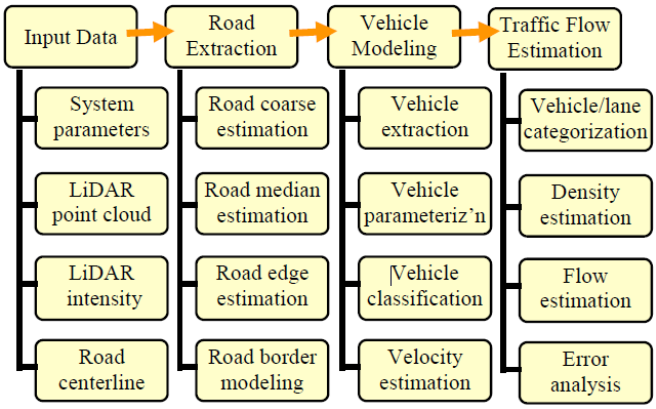


Figure 2.2 Data processing architecture.
Source: [16]

Zhao et al. [17], in 2006 study the primary application of a LiDAR at an intersection for tracking and classification of an object. They proposed a methodology that can classify the object into three different categories:

1) pedestrian, 2) bicycle, and 3) cars, buses, trucks. The object classification is done by Markov states in which an object model is predefined based on their typical appearances. The tracking of an object is done by matching frame to frame data generated at each scanning cycle of the LiDAR. The data is collected at an intersection using SICK LMS291 LiDAR with a video camera to compare the results. The results show that the developed methodology can attain 95% accuracy after comparing it with ground truth data. However, only ten minutes of collected data is processed for this study.

Yao [18] presents an algorithm to accurately track a vehicle using a LiDAR sensor through an intersection. The primary objective of the study is to collect stopping distance measurement of the vehicles at the intersection. A four-step

algorithm is introduced, which first identifies the object, extract the feature points of the detected object, track an object, and later calculate the stopping distance. The data are collected by installing a LiDAR sensor at the corner of the intersection. Object extraction with the threshold value is applied to identify the object from the background of the LiDAR data. The feature points are extracted to represent an entire cluster with a single position by fitting the front and side profiles of the vehicle with lines and finding the intersection of these two lines. Small gaps between the clusters of the feature points over time are analyzed to track the vehicles. The LiDAR is interfaced with a computer that runs a tracking algorithm to provide real-time estimates for the stopping distance. The algorithm proved to be effective in extracting vehicle objects, using a static background and a static threshold.

Fod et al. [19] propose a method to track people in crowded using single or multiple laser range finder data. A Kalman filter-based trajectory is proposed in this research. The proposed method performed under different scenarios with the SICK planner scanning laser finder. The results illustrate that the proposed method able to track multiple people with low errors and with reasonable computational efficiency.

Zhao and Shibasajki [20] study the application of LiDAR in indoor areas such as malls and exhibition halls to detect and track pedestrians using a single LiDAR scanner. The data are collected using a single-row laser range finder. The moving feet profiles are extracted from raw data and spatially integrated into a global coordinate system. A simplified Kalman filter is used to track pedestrian

trajectories. The method is evaluated using real-world and simulation-based data. The results suggest that the developed method is not robust and accurate under crowded place in the indoor environment compared to the crowded environment in an open area.

Cui et al. [21] propose an algorithm to track people using multiple LiDAR sensors in an indoor environment. From the obtained raw data, stable features are extracted, which are the movement of legs of people from successive laser frames. A tracker is developed based on Kalman Filter. The developed algorithm is tested in the indoor environment at the exhibition hall; the result shows that the method is robust and works well compare to conventional laser-based trackers such as measurement split and temporal occlusion.

Nashashibi et al. [22] introduce a robust method for detection, tracking, and classification of vehicles using mobile LiDAR sensors. The algorithm consists of three stages to detect and classify an object as a vehicle. The first stage uses the Ramer algorithm [23] to create a set of reflected data points using collected distance data. The Ramer algorithm allows reducing the number of points in a curve that is approximated to create a line segment. The second stage classifies the objects by analyzing length, vertices, and orientation of the points to the sensor. The third stage performs occlusion that handles the missing data causes due to obstructed LiDAR's field of view. The results show that the developed methodology is able to classify the objects as vehicles in the background of the noisy data.

Yang [24] develop a methodology to extract vehicles from the LiDAR

collected data. The system uses two LiDAR sensors mounted on the top of the probe vehicle to estimate the speed and length of the surrounding vehicles. Data from LiDAR sensors are first processed and combined to generate a 3-D LiDAR image. The methodology is efficiently extracted vehicles from the background environment with a miss-detection rate of 3.3% and a non-vehicle detection rate of 3.6%. Moreover, the vehicle classification system sorted vehicles into the predefined classes with 92.6% accuracy.

Thornton et al. [25] examine the applicability of two-dimensional mobile LiDAR sensors for parking system management. The pilot study is conducted by placing four LiDAR sensors on the top of the driver's cabin. Host vehicles are equipped with multiple GPS to locate the host vehicle's presence on the road network. The proposed methodology is managed the data in three different stages. The first stage considers the position of the probe vehicle using GPS data for each LiDAR scan. The second stage integrates the information from successive LiDAR scans to identify vehicles from the background. After accounting for occlusions, the algorithm reports the globally referenced location for each vehicle, as well as measuring the vehicle's length, height, and heading. The third stage uses the vehicle information and assigned them to road segments and lanes, reporting the occupancy of each parking zone, along with other measures such as a vehicle-to-vehicle gap. The proposed algorithm results are compared with the ground truth data collected through video and results showed 15 miscounts with a yielding error rate of 0.8%.

Zhang et al. [26] suggest a LiDAR-based vehicle detection approach by

utilizing its range of information. The authors introduce a Difference of Normals (DoN) method [27] for potential clustering of 3-D LiDAR points cloud into a vehicle, pedestrian, bicycle, and streetlight lamp. Moreover, the Support Vector Machine (SVM) [28] is used to classify the clusters as vehicles and non-vehicle objects. The evaluation results illustrated 89.3% accuracy in classifying vehicles in urban environments.

Thuy and León [29] propose a method to track an object over time using 2-D LiDAR. The researchers introduce a multi-modal object-tracking algorithm that employs a particle filter-based solution. The study's primary objective is to eliminate the error propagation that occurred in the tracking due to the use of a linear Kalman filter. The study uses particle filter-based Monte Carlo simulations for object tracking to model the non-linear process. The suggested method is applied to the data collected by two separate one-layer scanners (2-D LiDAR), which are synchronized with an angular resolution of a half degree and an overall angular range of 180 degrees. Data are collected by mounting one of the LiDAR on the front bumper and the second LiDAR on the rear bumper. The vehicle is equipped with the Inertial Measurement Unit (IMU) and combined with a DGPS to obtain the precise location of the vehicle and LiDAR. The result shows that the proposed method reduces the error while tracking an object in a 2-D LiDAR environment.

Taipalus and Ahtiainen [30] present an algorithm for detecting and tracking walking humans using 2-D mobile LiDAR. The algorithm consists of two separate steps; the first step identifies the detected cluster, and the second step tracks the

defined cluster over time within the scanning range of the LiDAR. A list of predefined features is provided in the process to identify the detected cluster points as a human. Based on the predefined features, if the two different clusters satisfy the condition, the object is defined as a human, and the human target is generated to track within the scanning range.

Tarko et al. [31] develop a traffic scanner (TScan) method to measure and collect traffic data at an intersection accurately. A 64 channel 3-D Velodyne HD-LiDAR is used to collect the traffic data. Distance information from the sensors is grouped and applied to spherical coordinates to identify the objects in the background. Once the object is identified, the clustering method is applied to define the collected points are from the same object or not. The clustering method analyzes the gap between the two successive LiDAR points; if the analyzed gap is higher than the threshold value, than it is assumed that the points were from the different physical surfaces. Once the detected object identified, a Kalman filter is applied to track the vehicle through the scanning range of the LiDAR. After tracking, the individual moving objects are classified into heavy and non-heavy vehicles, bicycles, and pedestrians. The results indicate that the method measured the vehicle positions and speeds with the higher accuracy.

Kluge et al. [32] present a method to track multiple objects using laser range finder data. The method consists of steps such as object identification, object extraction, object matching, and object tracking. Objects are extracted from the laser range finder by calculating the difference between the successive laser range obtained data; if the difference is higher than the threshold, then the obtained data

identified as an object. Object identification and extraction are made by segmenting the scanning data into different groups, and the threshold value is chosen to identify the maximum gap and classified as different objects. A graph theory [33] and bipartite graph [34] is used to correspond the object of one scan into the successive scans.

2.3 Continuous Wavelet Transform (CWT) Applications for Signal Analysis

CWT is one of the efficient tools that allow analyzing the signal in detail by breaking it down into small waves, which are highly localized in time. Over the period, CWT has been used in Civil and Transportation Engineering studies for various purposes, such as identifying the location of the cracks in the structure and location of the bottlenecks on road segment.

Ovanesova et al. [35] display the application of the discrete wavelet transform (DWT) to identify the cracks in different frame structures. The basic idea to use a wavelet transform for crack identification is to capture discontinuity in the signal since structure with a crack generates a difference signal compared to the crack-free structure. The proposed method does not depend on the previous knowledge of the undamaged structure to differentiate the signal caused due to crack in the structure. Moreover, the developed method can localize the source of the crack in the structures using the response signal. The study results indicate the effectiveness and responsiveness of wavelet analysis to extract the damage information from the response signal for various structures.

Rucka et al. [36] study the application of wavelet transforms for damage localization in the structure. The developed method is based on a two-dimensional continuous wavelet and applied on a cantilever beam and a plate with four fixed supports. The study finds that the CWT is more effective than the DWT.

Solis et al. [37] study the same application of wavelet transform for beams to locate the damage from changes in the mode shapes. The difference between the proposed method and the previous method is that the developed system requires the mode shape of the undamaged structure to differentiate it accurately. Moreover, in this study, the modal parameters and wavelet analysis are combined. The secondary objective of the study is to localize the section in the frame that can be affected by the damage. The result shows that the developed methodology is very sensible to capture the smallest crack in the beam.

Rakowski [38] discuss the application of the discrete dyadic wavelet transform (DDWT) to detect and localize the response signal features (slopes) caused by cracks in mechanical and civil structures. The primary objective of the study is to replace the computational burden of CWT to detect cracks that proposed in the previous research. The developed method is tested on the artificial signal of the length of 512 samples that consists of different slopes that originated from the damaged and undamaged structure. The results show that the method can detect the slightest change in the slope caused by a damaged structure.

Application of wavelet transform analysis in traffic engineering is not new; Jiang et al. [39] develop a wavelet function to study traffic flow pattern analysis. The main objective of the research is to develop an autocorrelation function (ACF)

that helps to select a decomposition level of wavelet analysis for time series analysis of traffic flow. A hybrid wavelet packet-ACF method is developed that identifies the unique and fractal properties of the traffic flow combined with the discrete wavelet packet transform (DWPT) based denoising technique. The results suggest that the proposed method can capture the essential characteristics of traffic flow that can help to build an accurate traffic forecasting model.

Giralda et al. [40] study the application of wavelet transform combined with neural networks for time series analysis of volume data. A fuzzy neural network with a stationary wavelet denoising process is introduced in the method to learn the time-series pattern of the traffic. The threshold values for the denoising process are determined by considering the fluctuation in the traffic flow. The developed methodology is applied to the data collected from three different interstate road and compared with the historical average, Kalman Filter, multilayer feed-forward neural network, and μ-support vector machine. The results suggest that the proposed model can achieve similar accuracy as the previous models.

Zheng et al. [41] propose the application of a wavelet transform to study the location of a bottleneck in congested conditions using loop detector data. The methodology considers the wavelet-based energy of speed signals to locate the actual bottleneck. A segment of US 101 in Los Angles, California, is used to study the effectiveness of the methodology. The results justify CWT's capability to identify the location of multiple bottlenecks and conclusively showed the origins of deceleration waves.

Mohan et al. [42] demonstrate the use of spectral graph wavelet transform to identify the traffic congestion on the road network by analyzing speed as an input signal. The method first calculates scaling functions using the spectral decomposition of the discrete graph Laplacian [43]. A wavelet coefficient is used to detect recurring or non-recurring congestion. The developed method is applied to a real-world expressway segment in Singapore. The results show that the method is able to detect speed changes during morning and evening peak hours. It is observed that the lower wavelet coefficients capture the minor changes in the signals.

Zhao et al. [44] propose a CWT based methodology to detect the shockwaves by investigating sudden drops in speed collected from Connected/Automated vehicles. The primary objective of the research study is to avert the propagation of the shockwave to upstream traffic. The proposed methodology is applied with VISSIM simulation; the results proved the effectiveness of CWT in detecting the shockwave by reducing travel time and delay by approximately 9% and 18%, respectively. Moreover, a developed methodology can detect the changes in the signal's energy obtained from CWT to identify the shockwave starting point.

2.4 Inverse Sensor Model

Inverse sensor model and grid occupancy analysis is one of the widely accepted computer programing algorithms that deal with the development of maps using measurement data with a known robot position on the map. Currently, grid

occupancy analysis is extensively used in connected and autonomous vehicles (CAV) applications.

The early application of the occupancy grid mapping is going back to the year 1988. Matthies and Elfes [45] propose the application of occupancy grid mapping fusing multiple sonar sensors for robot navigation. Bayesian estimation is applied at each scan to update the map of the surrounding area. The study uses the range information obtained from sonar and prepared a 2-D map. The probability of occupancy for each cell is calculated based on the collected measurement, and the Bayesian estimation scheme is applied to update the map recursively. The results show that the occupancy grid analysis is able to produce accurate surrounding maps using noisy data.

Vu et al. [46] present a simultaneous localization and mapping (SLAM) for detection and tracking an object in real-time. For the real-time application, the algorithm matches an incremental scan with odometry reading that works accurately in a dynamic environment. A global nearest neighborhood (GNN) is used to track the detected vehicle in real-time. The proposed method is tested under various conditions such as city streets, country roads, highways with maximum speed. The results suggest that the system can detect pedestrians and cars moving at a different speed and ability to generate accurate trajectories.

In recent years Bouzouraa and Hofmann [47] present a method to integrate occupancy grid mapping with model-based object tracking. A primary objective of the conducted research is to propose a universal framework to generate surrounding vehicle maps that can be used for future driving assistance systems.

The authors present a robust algorithm to detect dynamic obstacles from the occupancy map and accommodate the mapping process with the object-tracking algorithm in real-time. Results with real sensor data and a reference system show that the proposed method is robust and accurate to process the occupancy grid map and object-tracking algorithm simultaneously.

Wong et al. [48] propose a temporal joint probabilistic data association (JPDA) that combines the data from the previous scan, current scan, and next scan. The primary objective is to accurately detect and track an object that might be missed in consecutive scans since the previous research did not focus on that. The developed JPDA technique is associated with a SLAM algorithm. The technique is evaluated with an EKF based SLAM under two different scenarios in MATLAB based simulation. In the first scenario, an indoor environment is considered with static landmarks. The second scenario consists of static landmarks with a moving object, such as walking people. The results reveal that the method works well under scenario one, where there is no mobile object in the surrounding environment. On the other side, the method is not able to update the moving object accurately. The authors concluded that the method is more robust in the less dense area.

Moras et al. [49] improvise the occupancy grid mapping by dealing with the various source of uncertainty for autonomous vehicles. The uncertainties are caused due to the moving obstacles, and inaccurate pose estimation with noisy data. This study's primary objective is to remove clustering calculation, which is based on the assumption of the shape of objects. A first idea of the developed

method is to fuse the global grid map with a local grid map of two sensors. The developed method is tested with real-world data collected in an urban environment, and the results are compared with the previous approach. Experiment results suggest that the method is able to capture mobile objects with accuracy which helps to improve simultaneous localization using grid mapping.

Schütz et al. [50] propose a robust approach to estimate the object shape robustly while integrating free space information obtained from LiDAR sensors. An occupancy grid map is fused with the object tracking to generate more detailed shape and tracking information on the grid map. To estimate the dynamic and static shape of the object, a particle filter is used. The proposed method is evaluated with a 4-layer laser scanner with 25Hz rational frequency mounted on the front side of the vehicle. The results are evaluated with a ground truth system in the form of two DGPS systems, each fused with a highly accurate IMU. Results show that the proposed method improved both object tracking and shape estimation compared to standard tracking using the box model.

Hadji et al. [51] study the integration of the inverse sensor model with occupancy grid mapping. The primary objective is to eliminate the independency of nearby cells in occupancy grid mapping and consider the dependency of all cells in scanning. The primary objective of the research is to develop a framework to build a map autonomously with each scanning. The methodology is tested in an indoor environment by attaching LiDAR with a robot with the various surrounding environment. From the collected results, the authors could not draw accuracy since it required more data to collect.

Dia et al. [52] study further the integration of inverse sensor model and occupancy grid mapping. The main objective of the study is to understand the effect of sensor precision and grid resolution on the inverse sensor model. The method mainly focuses on single target sensors. The authors mathematically prove the strong relationship between the grid resolution and the sensor's precision on the occupancy estimations.

Kim et al. [53] propose a methodology for occupancy grid mapping with a deep neural network. The method uses a recurrent neural network called long-short term memory (LSTM) that study the temporal behavior and calculate future cell location of the detected object. The collected LiDAR information is sent to the LSTM, which calculates the probability of an object's future location on the grid. To evaluate the developed method, data is collected from the highway, and results are compared with Kalman filter-based occupancy grid mapping. The results suggest that the proposed system is accurate to predict the vehicle's future location on the grid and can create an accurate trajectory.

Weston et al. [54] develop a similar application of occupancy grid mapping with the deep neural network as of Kim et al. The primary objective of the study is to deal with noisy data obtained from LiDAR due to various weather conditions and occlusion. The developed neural network works as a self-supervised learning engine that uses partially labeled data generated by LiDAR. Five hours of data are collected to evaluate the developed method under a dynamic urban environment. The result proves the efficiency of the method by successfully segmenting real world into occupied and free space.

Steyer et al. [55] propose a dynamic grid mapping approach incorporating the static occupancy, dynamic occupancy, free space, and combined hypothesis. An adapted evidential filter is used in a dynamic grid map that consistently estimates and accumulate hypotheses and distinguishes static and dynamic occupancy. The evidential grid mapping is combined with a low-level particle filter tracking to estimate cell velocity distributions and predict dynamic occupancy of the grid map. Static occupancy is directly modeled in the grid map without requiring particles, increasing efficiency, and improving the static/dynamic classification due to the persistent map accumulation. Experimental results with real sensor data suggest that the methodology is useful in challenging scenarios with occlusions and dense traffic.

2.5 Kalman Filter based Object Tracking

The application of Kalman Filter to track an object is more prevalent in robotics and gaining more attention in mobility area with the rise of CAV) and unmanned aerial vehicles (UAV). Moreover, the use of Kalman filter is not only limited to physically detected objects; over the past few years, researchers are using the Kalman filter to track an object using a computer vision (CV) technique.

Prevost et al. [56] develop an extended Kalman filter-based methodology to estimate the state of a UAV detected object and predict its trajectory. The trajectory of an object is calculated using the defined motion model in the Kalman filter after defining the space of an object in the space. The developed method is tested in simulation, and results suggest that the proposed model can predict an object's trajectory in a dynamic environment.

Subramanian et al. [57] propose an algorithm to fuse multiple sensor information and develop a guidance system for an autonomous vehicle. The multiple sensors consist of vision sensors, laser sensors, inertial measurement units, and speed sensors. A fuzzy logic Kalman filter is the main component of a proposed algorithm. Fuzzy logic with Kalman filter is used since it can effectively control the divergence and update the parameter in the Kalman filter process. The method is applied to guide a tractor. The results suggest that the proposed multisensory fusion-based guides the vehicle accurately compare to individual sensor-based guidance with the Kalman filter.

Li et al. [58] develop an approach to deal with an occlusion problem and track multiple objects in real-time. The methodology consists of the Kalman filter motion model. To study the effectiveness of the method, the video data are collected in an indoor and outdoor environment with the movement of humans and vehicles. The results show that the method can detect and track multiple objects in real-time.

Salarpour et al. [59] develop a method to detect and track an object in the video by using region and feature-based tracking Kalman filter. The object from the video is detected using a background subtraction method. The primary objective of the developed method is to minimize the computation time and improve the feature-based tracking from the video that can deal with the occlusion problem.

Barrios et al. [60] study the application of Kalman filter with internal multiple model (IMM) and geographic information system (GIS). An internal multiple model

is a system that calculates the future location of a vehicle. The primary objective of the study is to reduce the error of IMM by fusing Kalman filter, GIS, and GPS data at the curvature of the road. The method is tested in the real world. From the results, the authors conclude that the methodology can accurately predict the location of a vehicle at curvature, but it can be improved further.

Li et al. [61] present a method to detect and track objects that are in the near field of vehicle. The proposed method uses the eight ultrasonic sensors attached linearly on the side of the vehicle. An unscented Kalman filter and extended kalam filter is designed to track the objects. The method consists of an empirical detection model developed by studying the shape of the objects obtained from single sensors and multiple sensors. An experimental study is conducted under two scenarios; in the first scenario, the ultrasonic sensors mounted on the side of a vehicle, as shown in Figure 2.3, the ultrasonic sound waves fire at the same time from all the sensors. In the second scenario, the sensors are configured to fire sound waves simultaneously. The field test is conducted in an open area where the vehicle is driven in a straight line at a speed of 5 km/hr. The results are compared with the advanced triangulation method (ATM) by studying root mean square error (RMSE) of the actual position of an object versus the predicted positions. The proposed methodology based on EKF and UKF provides better results compared to the ATM.

Figure 2.3 Sensor array system layout.
Source: [61]

Maalej et al. [62] develop a methodology to detect and track objects using 3-D LiDAR point cloud. The primary object of the study is to improve the perception of autonomous vehicles (AV). The methodology consists of EKF and a convolution neural network (CNN). A primary purpose of using CNN is to identify persons and vehicles from 3-D LiDAR points cloud. The initial results show that the developed methodology can recognize and track objects accurately.

Veeraraghavann and Papanikolopoulos [63] propose a methodology to track vehicles at an intersection using a camera-based system. The vehicles are identified using region segmentation and color analysis. A mean shift tracking algorithm is combined with an extended Kalman filter to track each vehicle frame by frame. The effectiveness of the methodology is studied using real-world data. The results show that the proposed method can effortlessly detect and track vehicles at an intersection.

Bas et al. [64] develop a methodology to detect and count vehicles from the video footage. The vehicles are detected using the background subtraction

method, and the Kalman filter is applied to track the detected vehicles. The background subtraction is done by modeling a background scene with a gaussian mixture model (GMM), and the changes in the background pixel colors are used to detect a vehicle in the frame. A constant velocity model is defined in a Kalman filter to track a detected vehicle. The developed method is studied under various scenarios, such as different light and different weather conditions. The results show that the method can able to detect and track vehicles under various conditions effectively.

Asvadi et al. [65] propose a methodology to track the objects fusing multiple sensors such as 3-D LiDAR, 2-D Red – Green – Blue (RGB) camera images, and an INS. The detection of an object is done by an adaptive color-based mean shift. A mean shift gradient estimation is applied to 3-D cloud points to localize the vehicle in the field of view, and to track an object, and a constant velocity model is defined to Kalman Filter. The proposed methodology is tested on the data set collected from the real world, and the results show the high-performance accuracy under different driving scenarios.

2.6 Summary

To this point, it is observed that in the previous research efforts, the LiDAR applications for traffic data collection are conducted under the controlled environment using expensive 3-D LiDAR sensors. Moreover, the application of LiDAR is mainly studied for autonomous vehicle applications with additional sensors such as image sensors, GPS, etc. Additionally, no research has been conducted that focuses on the turning movement counts at an intersection using a

low-cost 2-D LiDAR sensor. Based on the conducted literature review, it is noticed that a CWT is very sensible to capture minor changes in the signal, which is used in the proposed methodology. An inverse sensor model with occupancy grid mapping has been widely used in robotics and autonomous vehicle to localize the objects in the surrounding area. The literature review of Kalman filter application for object tracking indicates its efficiency to track moving objects, that is used to design turning movement count methodology.

CHAPTER 3

TRAFFIC DATA COLLECTION METHODOLOGY USING 2-D LIDAR MEASUREMENTS

3.1 Methodology for Traffic Counts

This section presents an in-depth discussion on the methodology of LiDAR-based traffic data collection and its primary components: 1) LiDAR sensor, 2) CWT, and 3) SVM.

LiDAR technology is highly used in the autonomous car industry. Its higher demand makes technology economical and easily accessible. To achieve one of the primary objectives of this study to develop a low-cost vehicle count methodology using advanced technology, the LiDAR is adopted as sensor technology. Moreover, its 360-degree visibility and accurate distance information make it more reliable compare to RADAR, which uses electromagnetic waves instead of light rays.

The obtained distance information from the LiDAR sensor is transformed into a signal for consecutive time intervals; CWT is employed to detect any deviation in it. Since the collected LiDAR data consists of noises that cause the deviation in the signal, the CWT is applied in this study to filter such ambiguous data and determined legitimate deviation in the signal caused by vehicles' presence. CWT allows analyzing the signal in detail by breaking it down into small waves, which are highly localized in time. CWT is able to detect small changes in the signal over time; the captured slight deviation in the signal advocates the presence of an object in the LiDAR scanning range.

SVM is one of the useful supervised machine learning tools for data classification and regression. In the methodology, the SVM is applied to classify the distance data points obtained from the sensor into detection and non-detection cases, which are highly complex. In the proposed methodology, a non-linear SVM with kernel function is used to classify the data points into two classes.

3.1.1 Basic Principle of LiDAR

Light Detection and Ranging (LIDAR) fires hundreds of thousands of laser pulses each second to the targeted object. A distance of the targeted/ detected object is calculated by considering a time pulse takes to return to the sensor from the object and with a known speed of light. Equation (3.1) represents the basic formula that determines the distance of the detected object in the vicinity of a LiDAR sensor.

$$Distance = \frac{Speed\ of\ Light * Time\ of\ Flight}{2} \tag{3.1}$$

LiDARs being used in the autonomous car industry, which is equipped with many emitters and receivers set, that allows it to produce a high sampling rate and a detailed point cloud of its surrounding environment. A single set of emitter and receiver (single channel) can process 100,000 pulses every second. Usually, high-end LiDARs are equipped with 16, 32, and 64 channels (multiple channels), which can generate millions of data points per second.

For the study purpose Scanse Sweep V1.0 LiDAR is used, which has an effective range of 40 meters (131 feet) with a maximum sample rate of 1,000

samples/second [68]. The Sweep LiDAR is a single plane scanner with a vertical field of view of 0.5 degrees. The LiDAR can be connected to microcontrollers or laptop using USB-to-serial converter [68].

3.1.2 Continuous Wavelet Transform (CWT)

This section explains a CWT and its application in the dissertation. Wavelet transform has gained the attention of mathematicians over the last couple of decades and very prominent tool in the field of engineering, image processing, and signal processing. Wavelet transform is an improvised version of Fourier transform, which allows analyzing nonstationary signals over time. The Wavelet Transform is defined in two categories considering the type of signal;

1) Continuous Wavelet Transform (CWT) and 2) Discrete Wavelet Transform (DWT). A CWT can be described by a continuous function, where DWT is a wavelet transform which is discretely sampled over time. A mathematical condition for wavelet is given as:

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt < \infty; \tag{3.2}$$

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt = 0; \tag{3.3}$$

$$\int_{-\infty}^{\infty} \frac{\left|\widehat{\Psi}(\omega)\right|^2}{|\omega|} d < 0; \tag{3.4}$$

Equation (3.4) is admissibility condition, where $\widehat{\Psi}(\omega)$ is the Fourier Transform of $\Psi(t)$. A wavelet function is constructed from translations and dilations of single function (mother wavelet) $\Psi(t)$ and defined as below,

$$\Psi_{\alpha,\beta}(t) = \frac{1}{\sqrt{|\alpha|}}\Psi\left(\frac{t-\beta}{\alpha}\right) \tag{3.5}$$

Where,

$\alpha, \beta \in R, and \ \alpha \neq 0$

The parameter α in Equation (3.5) called scaling parameter that measures the degree of compression of a signal. A value less than 1 indicates the compressed version of the mother wavelet caused due to high frequencies. A value greater than 1 for scaling parameter suggests lower frequencies of signal and $\Psi_{\alpha,\beta}(t)$ has a larger time width than $\Psi(t)$. The parameter β defined as a translation parameter regulates the time location of the wavelet.

If a function $f \in L_2(R)$, the series can be defined as below,

$$\sum_{\alpha\in R}\sum_{\beta\in R}\langle f, \Psi_{\alpha,\beta}\rangle\Psi_{\alpha,\beta}(t) \tag{3.6}$$

The Equation (3.6) can be called the wavelet series of f and $\langle f, \Psi_{\alpha,\beta}\rangle$ can describes as shown in Equation (3.7), which is a wavelet coefficient of f.

$$\langle f, \Psi_{\alpha,\beta} \rangle = d_{\alpha,\beta} = \int_{-\infty}^{\infty} f(t)\Psi_{\alpha,\beta}(t)dt \tag{3.7}$$

For CWT, the Equation (3.7) can be described as,

$$CWT_f(\alpha,\beta) = \langle f, \Psi_{\alpha,\beta} \rangle = d_{\alpha,\beta} = \int_{-\infty}^{\infty} f(t)\Psi_{\alpha,\beta}(t)dt \tag{3.8}$$

Where,

$$f(t) = \frac{1}{C_\Psi} \int_{-\infty}^{\infty} CWT_f(\alpha,\beta)\, \Psi_{\alpha,\beta}(t) \left(\frac{d\alpha d\beta}{\alpha^2}\right) \tag{3.9}$$

The general formulation of CWT for continuous signal is given as,

$$Q_{f,\Psi}(\alpha,\beta) = \int_{-\infty}^{\infty} f(t)(\alpha)^{-1/2}\Psi\left(\frac{t-\beta}{\alpha}\right)dt \tag{3.10}$$

Mother wavelets are characterized by properties such as orthogonality, compact support, symmetry, and vanishing moment. The Morlet and the Mexican hat are the two functions used for the wavelet analysis [66]. To select the wavelet function, the similarity between the signal and the mother wavelet is considered. In this study, a “Mexican hat wavelet” is selected for CWT analysis considering the resemblance of Mexican hat function’s shape with an input signal. The Mexican hat wavelet is defined as the second derivative of the Gaussian function:

$$\omega(t) = \frac{1}{\sqrt{2\pi} * \alpha} e^{\left(\frac{-t^2}{2*\alpha^2}\right)} \tag{3.11}$$

$$\Psi\left(\frac{t-\beta}{\alpha}\right) = \frac{2}{\sqrt{2\pi} * \alpha^3} \left(e^{\left(\frac{-t^2}{2*\alpha^2}\right)} * \left(\frac{t^2}{\alpha^2} - 1\right)\right) \tag{3.12}$$

The average wavelet energy,$E(\beta)$ at r is calculated based on CWT for different scales using Equation (3.13),

$$E(\beta) = \frac{1}{\max(\alpha)} \int_0^{\infty} \left|Q_f(\alpha, \beta)\right|^2 d\alpha \tag{3.13}$$

It is noticed that the distance information consists some noises that causes deviation in signal which can be resulted into false detection of vehicle as shown in Figure 3.1. A CWT is applied to handle such kind of inconsistency in the signal. The distance covered by a light pulse from the LiDAR sensor to the target surface (road surface) is consistent at any given time without the presence of an object (vehicles). Thus, a vehicle's presence can be recognized by parsing little inconsistency (i.e., drop in the distance signal and peak in the calculated energy) in the transformed signal and calculated energy as shown in Figure 3.2. Figure 3.2(a) displays a detection case of vehicle, and Figure 3.2(b) indicates a no-detection case in the LiDAR scanning area for one second of time interval.

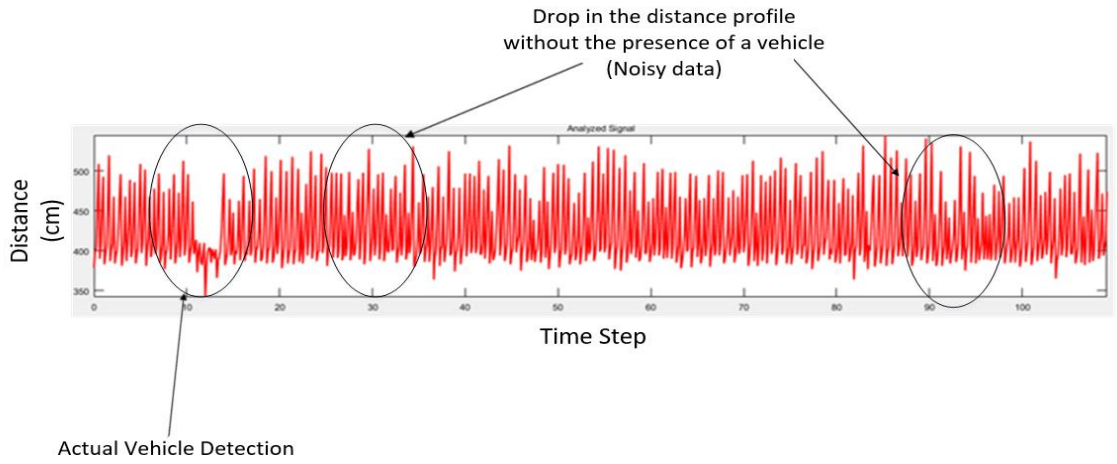
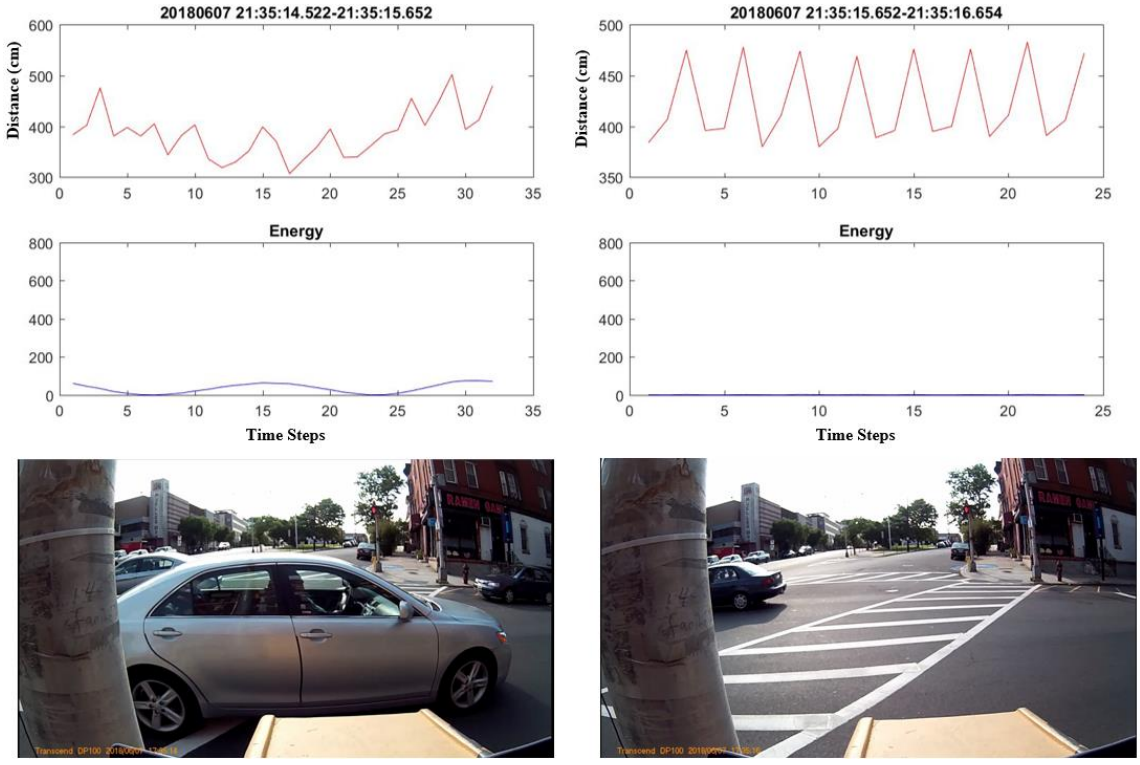


Figure 3.1 Converted distance profile to signal.



(a) **(b)**

Figure 3.2 Vehicle Detection Using Distance Profile Signal and Corresponding Energy.

3.1.3 Support Vector Machine (SVM)

In this section explanation of the Support Vector Machine (SVM) is given. The SVM is supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. An SVM can be applied for linearly and nonlinearly separable data sets. The SVM application on highly nonlinearly separable data is improved by using "kernel trick," which helps to project nonlinearly separable data into a higher dimension and then used that mapped data in a higher dimension to apply SVM. For the given set of training examples, the SVM constructs a model that assigns new examples into one category or other. For the classification, the SVM's primary function is to construct hyperplanes in a high or infinite-dimensional space. The best separation or classification of data sets is achieved when a hyperplane has the most considerable distance from the nearest training data point. Figure 3.3 shows the classification of the data by achieving maximum distance from hyperplane to the nearest data point. Following is the primary explanation of nonlinear SVM model.

For a given training data set, D,

$$D = \{(x_i, y_i)\} | x_i \in R^p, y_i \in \{-1,1\}_{i=1}^{n} \tag{3.14}$$

Where*:*

y_i = either 1or 0 which indicates the class to which the point x_i belongs.

Here, the objective is to identify maximum margin (C) hyperplane that divides the points from $y_i = 1$ to 0 as shown in Figure 3.3. The hyperplanes can be represented as follows:

$$W \cdot x - b = 1 \; and \; W \cdot x - b = 0 \tag{3.15}$$

Where:

W = Normal vector to the hyperplane

x = Set of points

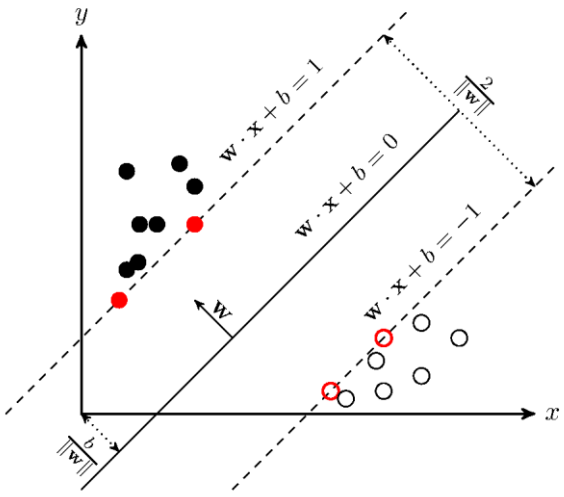


Figure 3.3 Data classification using SVM.

The parameter $b/\|W\|$ determines the offset of the hyperplane from the origin along the normal vector W. The distance between these two hyperplanes is $2/\|W\|$. Therefore, to get the maximum distance between two hyperplanes, the value of

$\|W\|$ should be minimize during the training process by considering the following constraint:

$$y_i(W \cdot x_i) - b \geq 1, for\ all\ 1 \leq i \leq n \tag{3.16}$$

In nonlinear classifier SVM, every dot product is replaced by the kernel function. That helps the algorithm to fit the maximum-margin hyperplane in transformed feature space. This transformation may be nonlinear and transformed in high dimensional space. There are different types of "kernel function/trick" such as polynomial, radial basis function, and hyperbolic tangent.

For analysis, a nonlinear SVM is selected to evaluate the calculated energy data as detection and non-detection cases since the energy data are closely located and not linearly separable. A Radial Basis Function (RBF) based kernel is used for SVM application. The RBF function transforms the nonlinearly separable data into a higher dimension, which helps to find optimal hyperplane width between two data sets. The RBF function is represented by Equation (3.17).

$$K(X, X') = \exp\left(-\frac{\|X - X'\|^2}{2 \cdot \sigma^2}\right) \tag{3.17}$$

Where,

σ is a kernel parameter.

The effectiveness of SVM is depends on kernel parameter (σ) and soft margin parameter (C). The best combination is selected using Bayesian optimization.

3.2 Methodology for Turning Movement Counts

This section discusses about the components used to develop vehicles' trajectory using 2-D LiDAR data. The proposed methodology consists of 1) Inverse Sensor Model, 2) K-Means Clustering, and 3) Kalman Filter.

3.2.1 Inverse Sensor Model

The inverse sensor model is often used in robotics to generate the surrounding map using range information collected by LiDAR or RADAR with the known position of the robot. An inverse sensor model is primarily defined as a state model for occupancy grid mapping. The state model is the map of the surrounding area that identified the location of the detected objects by converting polar coordinates to cartesian coordinates. A 2-D grid map is represented as shown in Figure 3.4.

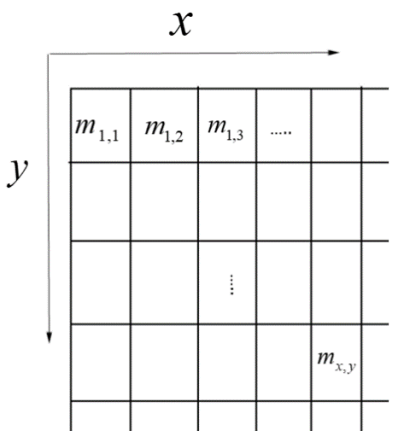


Figure 3.4 2-D Grid map.

In the state model, the distance measurement from the known LiDAR position is used to identify the exact coordinates of the objects on the grid map. Figure 3.5 represents a hypothetical grid map with a LiDAR. A cell highlighted in black represents the location of an object calculated using Equation (3.18) and (3.19). A Known state of the LiDAR is given by (x_1, x_2, θ).

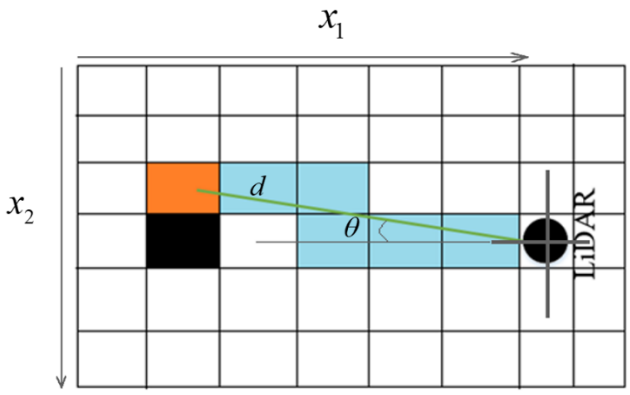


Figure 3.5 Grid cell detection at different azimuths.

$$\begin{bmatrix} x_{1,occ} \\ x_{2,occ} \end{bmatrix} = \begin{bmatrix} \cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.18)$$

$$\begin{bmatrix} i_{1,occ} \\ i_{2,occ} \end{bmatrix} = ceil\left(\frac{1}{r}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) \tag{3.19}$$

Since the LiDAR is generating multiple light beams and collect distance measurements at the different azimuths (angles) the Equations (3.20) and (3.21) can be written as;

$$\begin{bmatrix} x_{1k} \\ x_{2k,} \end{bmatrix} = \begin{bmatrix} d_k \cos(\theta + \alpha_k) \\ d_k \sin(\theta + \alpha_k) \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.20}$$

$$\begin{bmatrix} i_{1,occ} \\ i_{2,occ} \end{bmatrix} = ceil\left(\frac{1}{r}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) \tag{3.21}$$

Where,

Distance measurements: $d_k = (d_1, d_2, d_3, d_4, d_5)$

Direction of rays (azimuth): $\alpha_k = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$

3.2.2 K- Means Clustering

K-Means clustering is one of the unsupervised machine learning techniques to cluster the different data points into K clusters by calculating the nearest means. An unsupervised machine learning is selected since it can define the datasets into different groups without any known label, unlike supervised machine learning techniques. The K-Means clustering method uses an iterative process to achieve minimum distance between the centroid of the “K” groups and assigned data points

to that group. The less variation within clusters, the more homogeneous (similar) the data points are within the same cluster.

Equations from (3.22) to (3.24) explains the K-Means Clustering algorithm. A set of data $\{x_1, x_2, x_{3,\ldots\ldots} x_n\}$ s defined as an input to the K-mean clustering, where the defined data set is a d-dimensional data. The algorithm's primary objective is to assign the input data into the "k" cluster by minimizing the Euclidean distance between each set of data points and the centroid of the cluster. The objective function is defined as below,

$$S = argmin \sum_{k=1}^{k} \sum_{j=1}^{n} \omega_{jk} \left\| x^j - \mu_k \right\|^2 \tag{3.22}$$

Where,

$\omega_{jk} = 1$ for data points x^j if it belongs to cluster k else $\omega_{jk} = 0$

μ_k is the centroid of cluster x^j

The minimization process is conducted in two parts. First the Equation (3.22) is minimized with respect to ω_{jk} and fixed μ_k. During the minimization process, the function S is differentiated with respect to ω_{jk} and update the cluster assignments as shown in Equation (3.23). Second the function S is differentiated with respect to μ_k and compute the centroids after the cluster assignments from previous step as described in Equation (3.24).

$$\frac{\partial s}{\partial \omega_{jk}} = \sum_{k=1}^{k}\sum_{j=1}^{n}\left\|x^j - \mu_k\right\|^2 \Rightarrow \begin{cases} 1 \;\; if \;\; k = argmin_j\left\|x^j - \mu_k\right\|^2 \\ 0 \quad\quad otherwise \end{cases} \tag{3.23}$$

$$\frac{\partial s}{\partial \mu_k} = 2\sum_{j=1}^{n}\omega_{jk}\left(x^j - \mu_k\right) = 0 \Rightarrow \left\{\mu_k = \frac{\sum_{j=1}^{n}\omega_{jk}x^j}{\sum_{j=1}^{m}\omega_{jk}}\right. \tag{3.24}$$

In the proposed methodology, a K-Means Clustering is used to identify presence of multiple objects in the LiDAR field of view as shown in Figure 3.6. Figure 3.7 represents a logic about the K-means clustering steps, and a pseudo-code is provided in Appendix A.

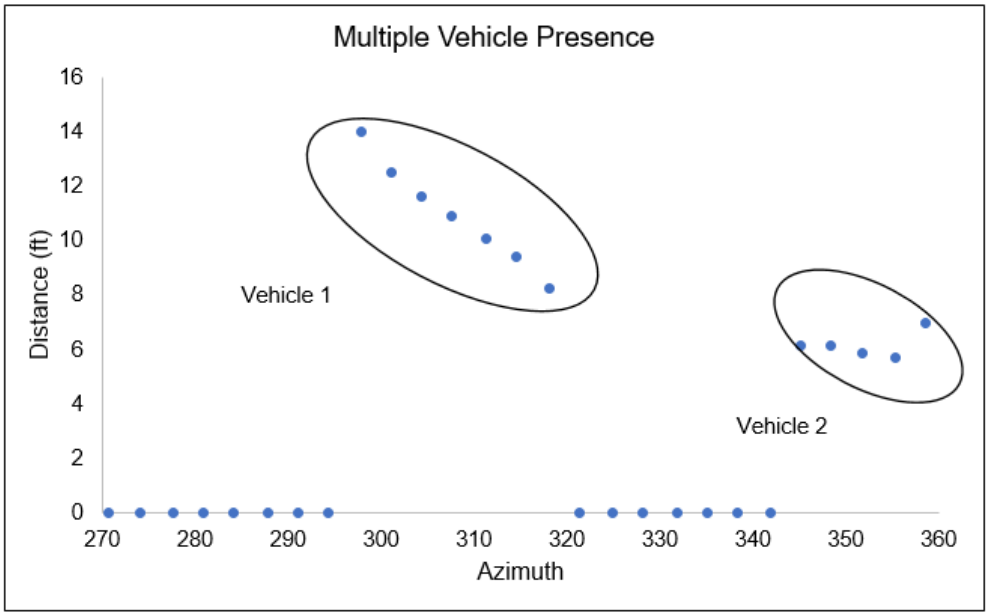


Figure 3.6 Multiple vehicle presence in LiDAR Field of View (FOV).

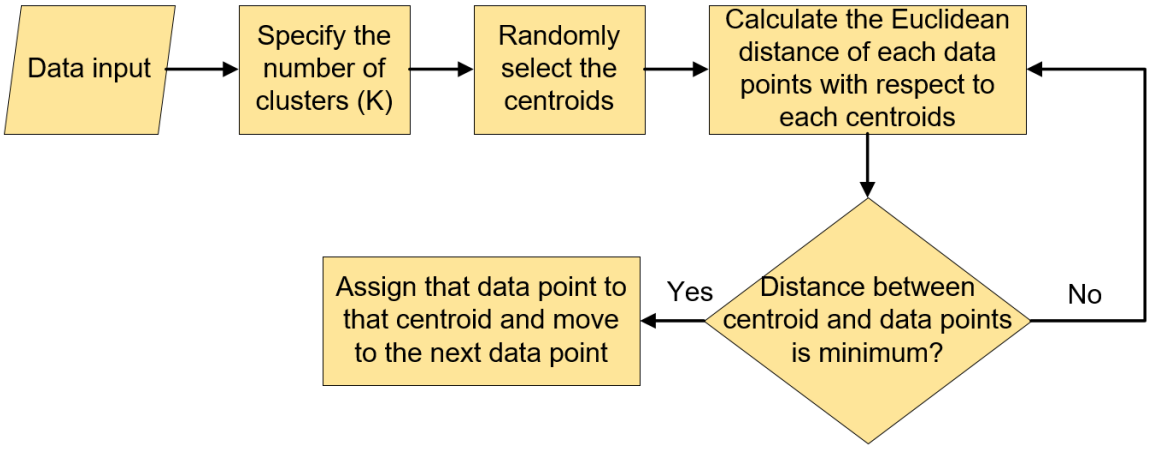


Figure 3.7 K-Means clustering flowchart.

3.2.3 Kalman Filter

Kalman filter is one of the popular approaches that has been used to estimate the state of the dynamic system over the period, and in this study, it is used to track the movement of detected vehicles. Furthermore, the Kalman Filter application is gaining more popularity as one of the noise cancelation techniques in sensor data. Since its introduction to the mathematics, the Kalman Filter has been updated for different applications. Kalman Filter is classified into three different types:

- A Discrete Kalman Filter, which is applied to solve a linear system problem.
- An Extended Kalman Filter, which is designed for the nonlinear system.
- An Unscented Kalman Filter applied for nonlinear function estimation and filtration.

In this study, a Discrete Kalman Filter is used for prediction and tracking purposes. Furthermore, it helps to deal with noisy data. During the application of an algorithm, the state is assumed to be a linear system with a gaussian distribution. A discrete Kalman filter consists of two main steps: 1) Prediction, and 2) Correction as shown in Figure 3.8. A prediction step allows estimating the

current state and error covariance to obtain the priory estimates for the next time step. The correction step is responsible for feedback by incorporating a new measurement into the priori estimate to get an improved estimation. Equations (3.25) and (3.26) represent the state prediction steps of discrete Kalman filter.

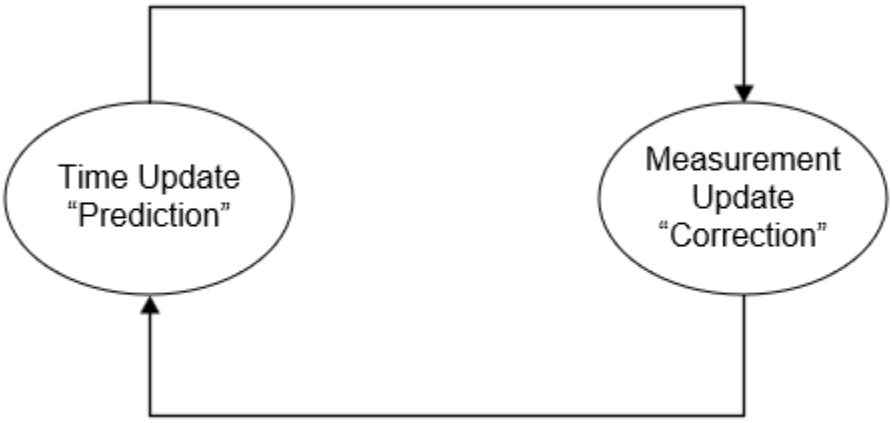


Figure 3.8 Discrete Kalman Filter cycle.

A constant velocity model is used for the Kalman filter, which is a renowned model to track moving objects. The model assumes that the velocity of an object is constant throughout the sampling interval.

$$\dot{\hat{X}}_s = A * \hat{X}_{s-1} + B * u_s + w_s \tag{3.25}$$

$$\dot{P}_s = A * P_{s-1} * A^T + Q_s \tag{3.26}$$

Where,

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

$$B = \begin{bmatrix} \frac{(\Delta t)^2}{2} & 0 \\ 0 & \frac{(\Delta t)^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{3.28}$$

$$u = \begin{bmatrix} 4 \\ 4 \\ 1 \\ 1 \end{bmatrix} \tag{3.29}$$

u = control variable of matrix

w = predicted state noise matrix

Q = process noise covariance matrix

X = state matrix

s = current step

$s - 1$ = previous step

P = state covariance matrix

A control variable in Equation (3.25) is an acceleration parameter of a vehicle at an intersection. Since the constant velocity model is defined for the Kalman filter, the acceleration of a vehicle is assumed constant with the value of 4

ft/s (1.22 m/s2) [67]. In Equation (3.26), Q preserves the state covariance matrix to become too small or zero and it is represented as shown in Equation (3.30).

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \Delta t & 0 & 1 & 0 \\ 0 & \Delta t & 0 & 1 \end{bmatrix} \tag{3.30}$$

Here, Δt represents the time difference between consecutive LiDAR scanning cycle. Equations (3.31) to (3.34) explains the measurement update steps. Equation (3.32) represents the update with the measurement by incorporating Kalman gain (K). A Kalman gain is represented as per Equation (3.31), which is a weight factor based on comparing the error in the estimate to the error in measurement. Equation (3.34) calculates a posterior error covariance. At the end of each update measurement step the process is repeated with posterior estimates to predict a new prior estimate. Figure 3.9 and 3.10 shows the detailed representation about the Discrete Kalman Filter algorithm.

$$K_s = \left(\dot{P}_s * H^T\right) * \left[\left(H * \dot{P}_s * H^T\right) + R\right]^{-1} \tag{3.31}$$

$$\widehat{X_s} = \hat{X}_s + \dot{[}K_s * (Y_s - H * \hat{\dot{X}}_s)] \tag{3.32}$$

Where,

R = sensor noise covariance matrix

Y_s = measurement input

$$Y_s = (H * X_s) + m_s \tag{3.33}$$

$$P_s = (I - K_s * H) * \dot{P}_s \tag{3.34}$$

$$R = \begin{bmatrix} (0.6542)^2 & 0 & 0 & 0 \\ 0 & (0.6542)^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}$$

A sensor noise covariance matrix R is represented as shown in Equation (3.35). Sensor noises often occur during transmitting and receiving the signals caused by either faulty communication or power supply. Data imputation is adopted to study the measurement noises of the LiDAR sensor during the data collection procedure. Random scanning cycles (pair of azimuth & distance) are selected from the data sets, which has no missing values. To apply data imputation, some distance information is randomly removed and predicted using data imputation. A linear interpolation-based imputation technique is applied. Calculated standard deviation is compared with the manufacturers' defined measurement error. Equation (3.36) represents the basic idea of linear regression. The observed standard deviation after data imputation is 0.652 feet (19.89 cm). The error in measurements provided by the manufacturer is 0.591 feet (18 cm), which is nearly equal to the calculated standard deviation in data imputation.

$$y - y_i = \frac{y_{i+n} - y_i}{x_{i+n} - x_i}(x - x_i) \tag{3.36}$$

Where,

$x =$ Initial Azimuth Value

$y =$ Missing distance information

$x_i =$ Incremental Azimuth values

$y_i =$ First nonzero distance information in vector

$y_{i+n} =$ Last nonzero distance information in vector

$x_{i+n} =$Last Azimuth Value

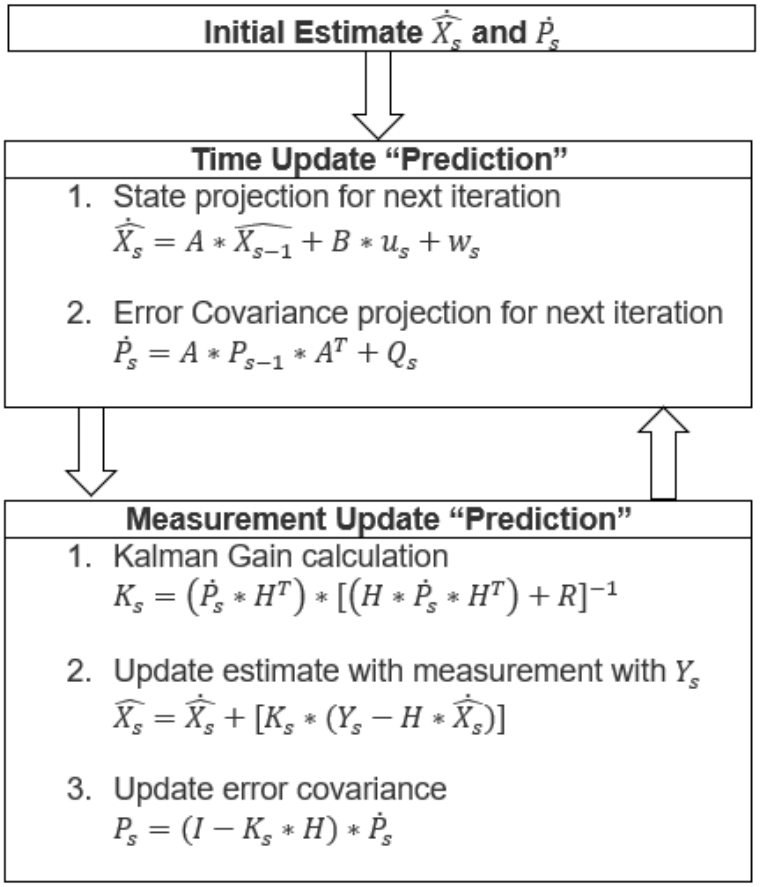


Figure 3.9 Discrete Kalman Filter high level diagram.

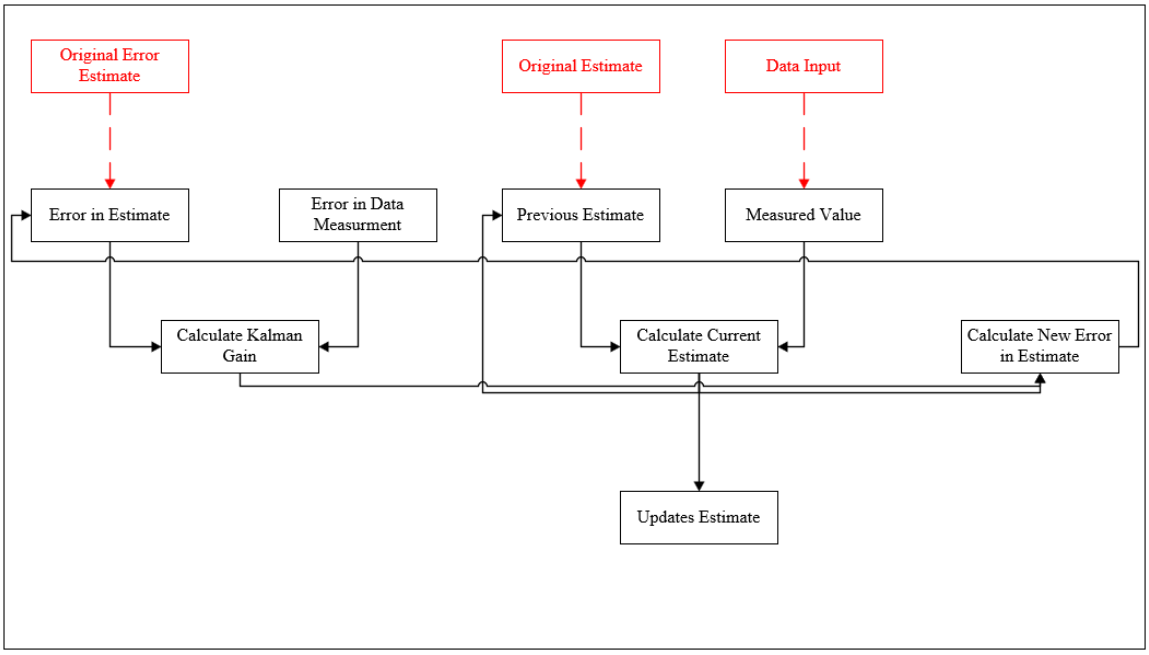


Figure 3.10 Discrete Kalman Filter essential flow chart.

CHAPTER 4

FRAMEWORK DEVELOPMENT

4.1 LiDAR Application for Traffic Counts

Figure 4.1 illustrates a high-level framework of the proposed methodology performing the following stepwise procedure to extract traffic count data from a LiDAR sensor: 1) LiDAR data collection;2) CWT application for LiDAR signal analysis, and 3) SVM application for data classification.

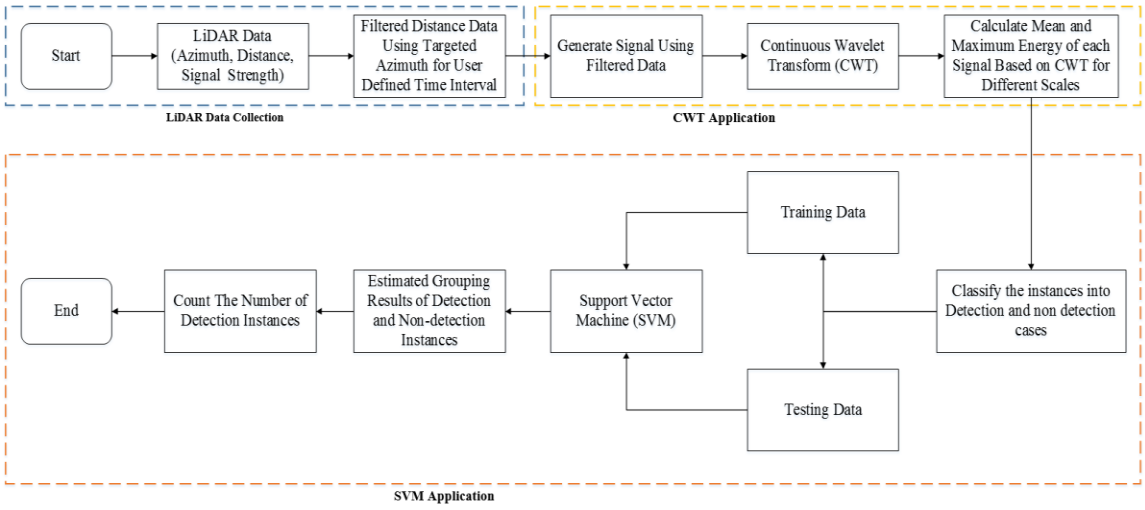


Figure 4.1 Proposed methodology for traffic counts using 2 - D LiDAR.

During the data collection process, a LiDAR sensor is placed on a traffic signal pole at such a height that it does not affect the ongoing traffic, as shown in Figure 4.2. The installed sensor scanned its surrounding area from horizontal to the vertical plane. The distance and signal strength information is obtained with the corresponding azimuth. The azimuth is the angle at which the sensor receives the reflected light rays. The distance profile (cm) from the LiDAR sensor to the road

surface is obtained for several scanning cycles. Here, the scanning cycle is one complete revolution of the LiDAR head. Since the LiDAR has 360-degree sensing capability, the next step separates the distance profile for user-defined azimuth, which gives the collected distance information within the sensor coverage area.

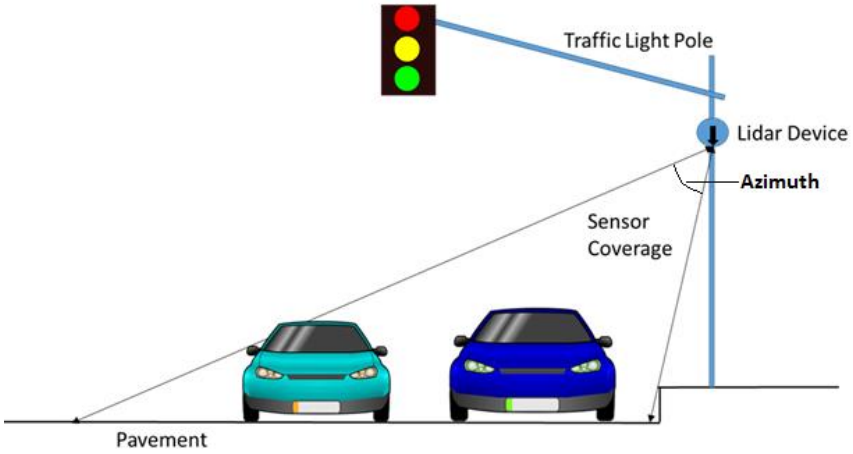


Figure 4.2 Visual presentation of LiDAR placement for traffic count data collection.

In this study, the distance profile for the pre-defined time interval is converted into a signal. The CWT is performed on the transformed signal to recognize disparities in it by using Equation (3.4). The average-based energy is then calculated for various scales by applying Equation (3.10) for each signal. The mean energy and maximum energy are calculated for each signal. Based on the mean and maximum energy, 50% of the data is classified into detection and non-detection cases by comparing it with collected videos from the field.

Using the classified data, the SVM is trained to determine the class of undisclosed distance profiles. The undisclosed data set is considered as a testing data set (50%) in this study. Once the SVM conducted to estimate the class of new

data points, the output is compared with the ground truth count to measure the accuracy of the proposed approach.

4.2 LiDAR Application for Turning Movement Counts

This chapter discusses a proposed methodology to construct vehicle trajectory and conduct the turning movement counts. Figure 4.3 represents an outline of the framework of the proposed approach. An algorithm is divided into three different stages. The grouping stage handles the data, and the detected object/s are identified as a vehicle or pedestrian using the K-means clustering technique. In the next stage, the detected object/s are localized using the inverse sensor model followed by the tracking stage using a Kalman filter.

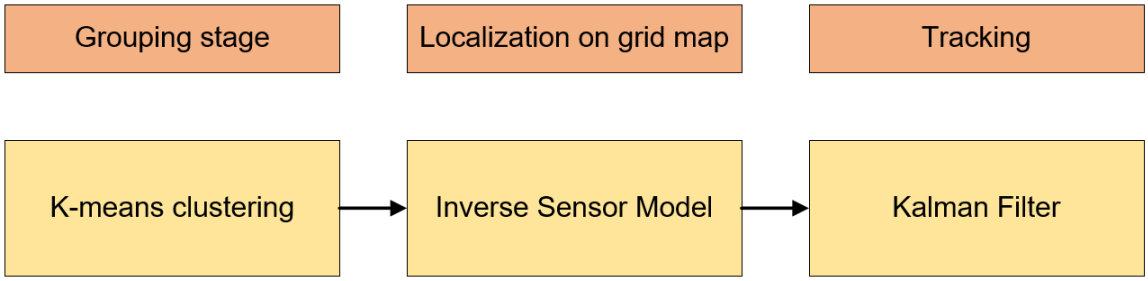


Figure 4.3 Outline of proposed methodology for turning movement counts.

At the beginning of the first stage, an empty matrix is created to save the updated input information. As an input, only azimuth (α) with respect to the field of view (FOV) is provided with its corresponding distance (cm) information. The Field of View (FOV) is defined as $0^0 <> 90^0, 90^0 <> 180^0, 180^0 <> 270^0, or\ 270^0 <> 360^0$ for an individual LiDAR. Since the grid map cell dimensions are defined in feet and make the calculation uncomplicated, the collected distance information is converted into feet from centimeter. From the input information, the first scanning

cycle is selected to check the distance information. Figure 4.4 shows the grouping stage algorithm's detailed information, which is repeated for each scanning cycle. The steps shown in Figure 4.4 is repeated until there is no detection for thirty scanning cycles. It is observed that an individual pedestrian generates less than two reference points.

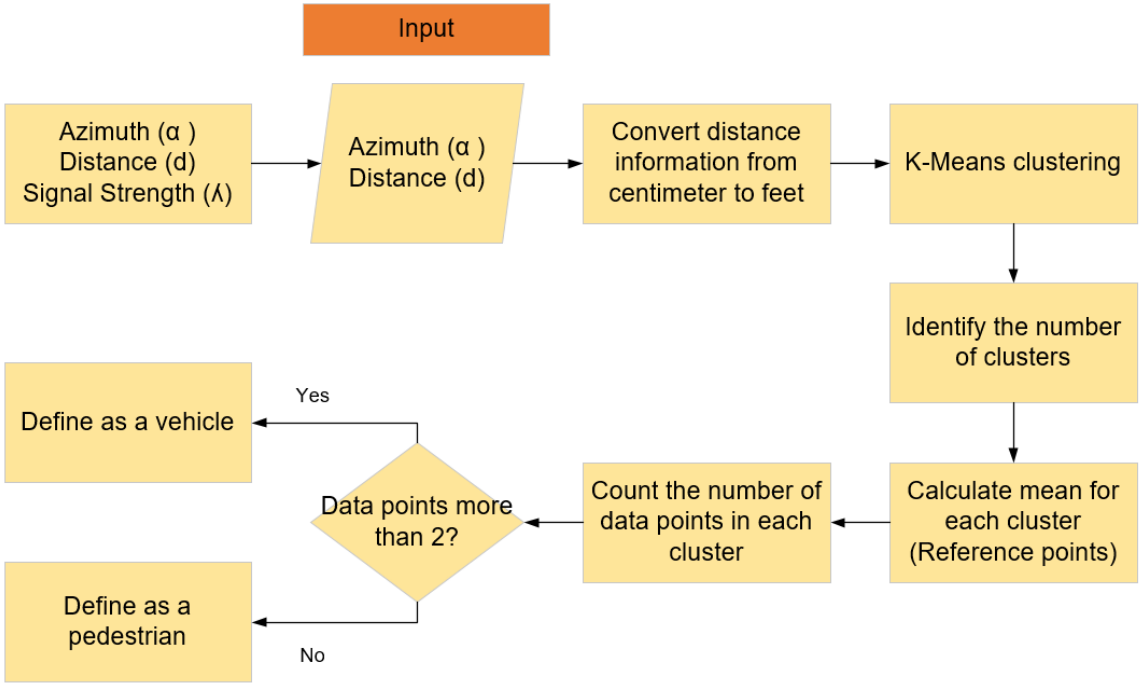


Figure 4.4 Flow chart of grouping stage.

Following Figure 4.5 shows the details to localize the detected vehicles on a grid map. The calculated mean values of the cluster from each scanning cycle is used as input data to localize the vehicles.

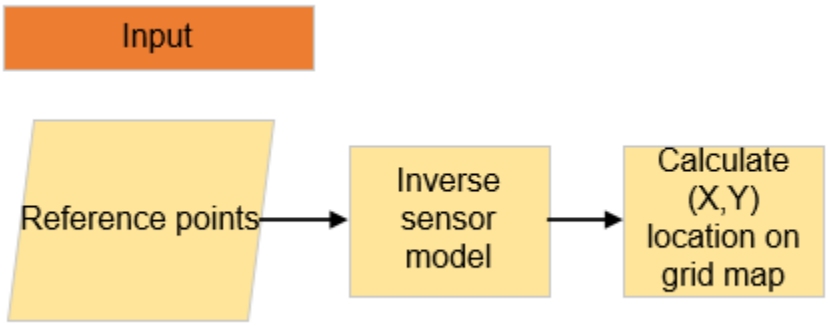
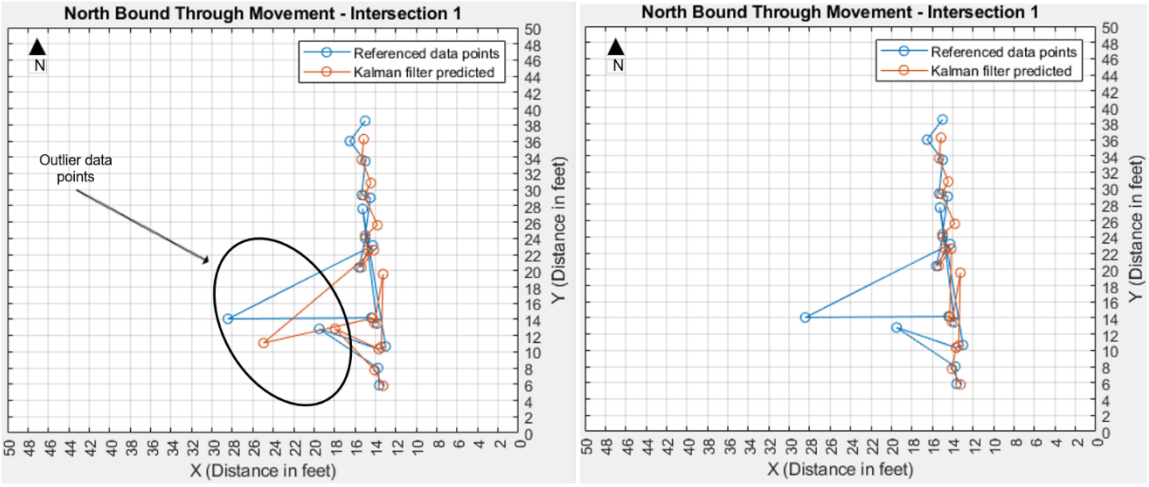


Figure 4.5 Flow chart of inverse sensor model.

In the tracking stage, a detected vehicle's trajectory is constructed with the application of the Kalman filter algorithm using calculated (X, Y) values. Figure 4.7 shows a detailed procedure to construct a vehicle's trajectory and define the movement on a grid map. It is noticed that due to some outlier data points, the obtained through trajectories of the detected vehicle are not precise, as shown in Figure 4.6. A threshold window is defined to remove the outlier data points. The threshold window is for any movement is calculated by ± 2 along its average value of primary axis. The obtained trajectory after removing the outlier data points is shown in Figure 4.6 (b). Figure 4.6 represents a North Bound through and right turning movements of the vehicles, respectively.

The movement of a vehicle is identified by calculating the variations of the plotted trajectory's data point along the primary axis in this study; axis perpendicular to the entrance point of the vehicle into the intersection is considered as a primary axis as shown in Figure 5.5 and 5.6. It is observed that the range of variation for through movement varies between 0.2 to 0.9. The small number of variations is caused due to minimal lateral movement of the vehicles. The

observed range of variation for non-through movements are between 1 to 11 caused due to significant lateral movement. The vehicle movement progression is considered in the field of view of LiDAR to define the turning movement of the vehicles as a left or right. Since the various azimuth windows are defined as the field of view of LiDAR, it is pronounced that the vehicle's progression either away or into the field of view can be used to define the right or left-turning movement.



a) Before application of threshold window

b) After application of threshold window

Figure 4.6 North bound through movement at an intersection 1.

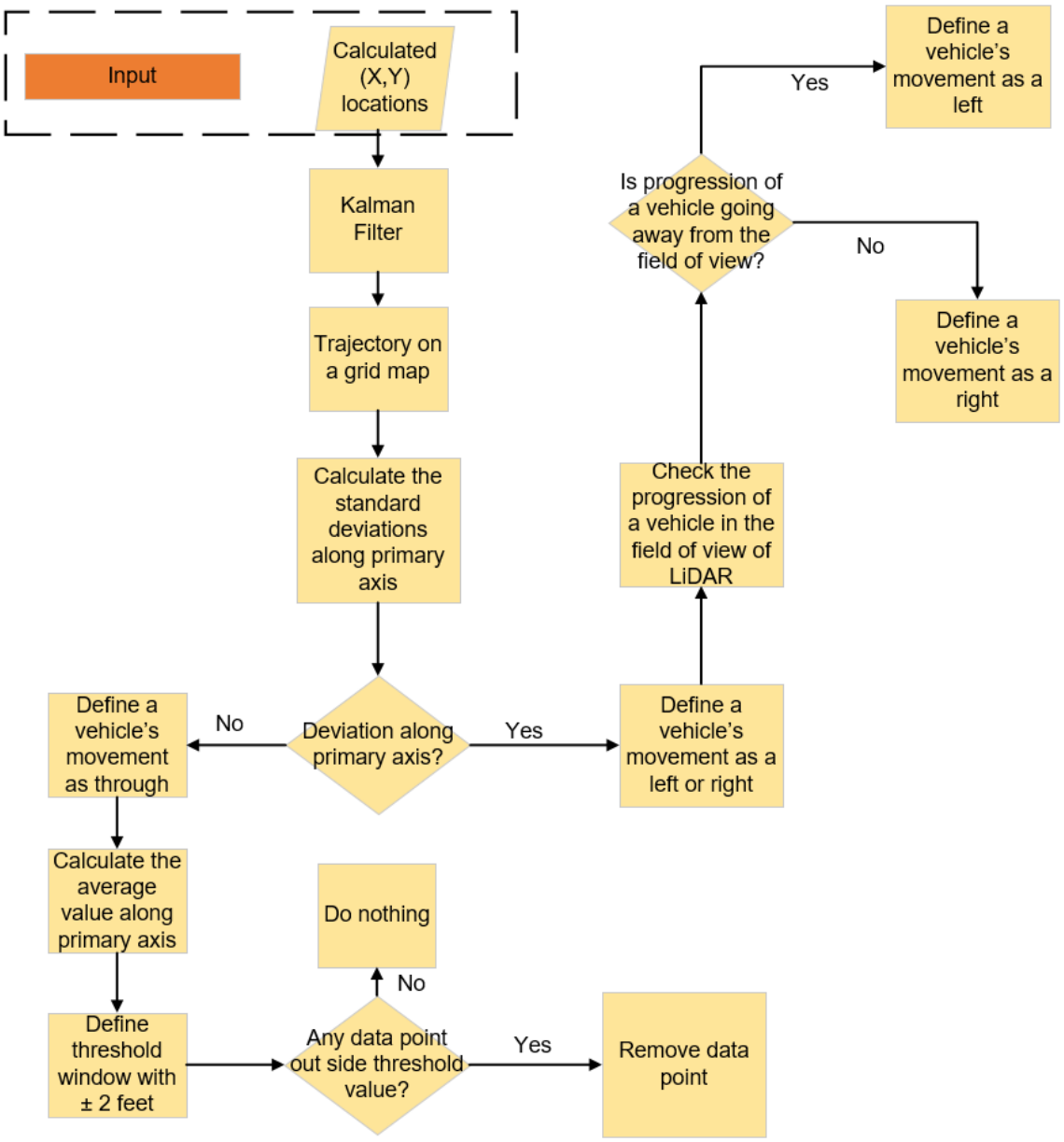


Figure 4.7 Flow chart of turning movements identification and Kalman Filter application.

CHAPTER 5

CASE STUDIES

This chapter presents case studies to examine the effectiveness of the proposed methodology. Data from three different intersections are collected for LiDAR application of traffic count, and two intersections are studied to conduct a turning movement count using LiDAR.

5.1 Data Collection for Traffic Count

For the proof of concept test, a Scanse Sweep LiDAR (68) sensor is used. The power is provided using a laptop through a USB-to-serial convertor. Data collection is conducted at three locations across the city of Newark, NJ. Figure 5.1 (a) shows a Scanse LiDAR sensor head and the installed LiDAR sensor at sites 1 and 2, respectively, in Figure 5.1 (b) and (c).

a) Scanse LiDAR b) Location 1 c) Location 2

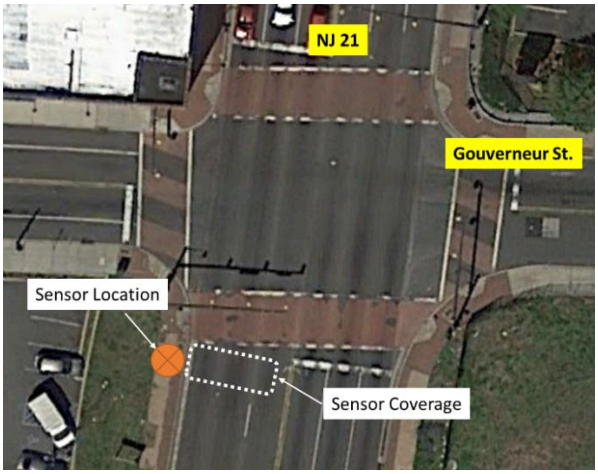
Figure 5.1 Scanse LiDAR head and lidar placements at location 1 & 2.

Table 5.1 shows the details of the test locations with data collection periods and sensor configurations. To collect high volume, the data collection is conducted during the peak hours at each location. The detection range of a LiDAR sensor is considered during the data collection to identify the number of lanes covered in the coverage area. A LiDAR sensor is installed on the roadside infrastructure (e.g., streetlight pole, traffic signal arm past) at least 13 feet above the ground.

Table 5.1 Data Collection Locations for Traffic Counts and LiDAR Configurations

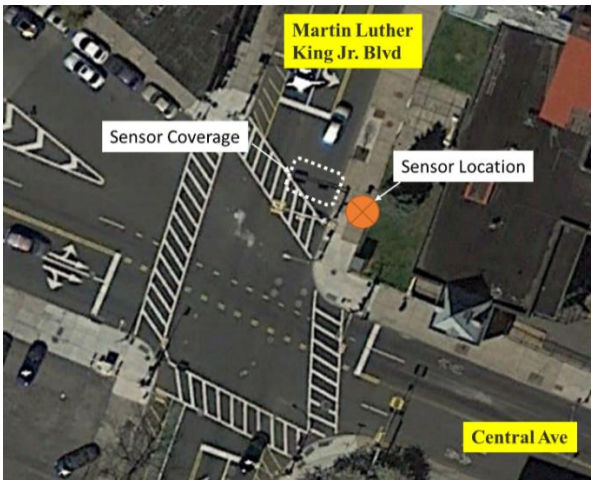
Location	Intersections	Time period	Rotation Frequency (Hz)	Sample Rate (Hz)	Studied Direction
1	Gouverneur St. at NJ-21	10:00 to 11:00	9	1000	South Bound
2	Irvine Turner Ave. at Muhammad Ali Ave.	17:36 to 18:24	8	1000	North Bound
3	Dr. Martin Luther King Jr. Blvd at Central Ave	17:37 to 18:11	8	1000	North Bound

Figure 5.2 shows the sensor locations deployed at each intersection with the coverage area. At location 1, only two out of three lanes traveling southbound are considered for data analysis considering a LiDAR's effective range. A total of 1-hour of data is collected at this location. The second location consists of one lane in each direction, and southbound traffic is captured for 48 minutes. For the third location, the traffic data for one lane traveling northbound direction is collected for 34 minutes.



(a) Location 1: Gouverneur St. at NJ 21

(b) Location 2: Irvine Turner Ave. at Muhammad Ali Ave



(c) Location 3: Dr. Martin Luther King Jr. Blvd at Central Ave.

Figure 5.2 LiDAR sensor locations for traffic counts data collection.

5.2 Data Collection for Turning Movement Counts

The data are collected from two different intersections located in Newark, NJ, to study the accuracy of the developed algorithm. Table 5.2 and 5.3 provides detailed information about the data collection locations. Figure 5.3 shows the graphical representation of the LiDAR placements at an intersection. The LiDAR sensors are installed at 3’ to 3.5’ from the ground to get enough reflection from the surface of the vehicles. At each intersection minimum of three LiDAR sensors are installed. The primary reason to deploy multiple sensors at the intersections is to deal with occlusion problems. The occlusion often caused by two vehicles traveling side by side or detection of a vehicle obstructed by pedestrians. Enhancement in the detection is the second reason to place multiple sensors since it is observed that the detection ability of Scanse Sweep LiDAR in an open area is reduced while scanning a horizontal plane.

Table 5.2 Data Collection Locations for Turning Movement Counts

Intersection Number	Intersection Name	Exclusive Left Turn	Exclusive Right Turn
1	Central Ave. & Lock St.	Yes (NB)	NA
2	Warren St. & Dr. Martin Luther King Jr. BLVD	NA	NA

A python-based program is used to automate the data collection process, which allows the collection of the data for a more extended time period without any external interruption. The raspberry pi minicomputers are used to run a python script and saving collected data. Figure 5.4 shows the field installation of LiDAR at intersection 1.

Table 5.3 Data Collection Time Period and LiDAR Configurations for Turning Movement Counts

Intersection Number	Intersection Name	Time period	Rotation Frequency (Hz)	Sample Rate (Hz)	Number of LiDAR
1	Central Ave. & Lock St.	16:23 to 18:17	10	1000	3
2	Warren St. & Dr. MLK BLVD	11:26 to 12:54	10	1000	4

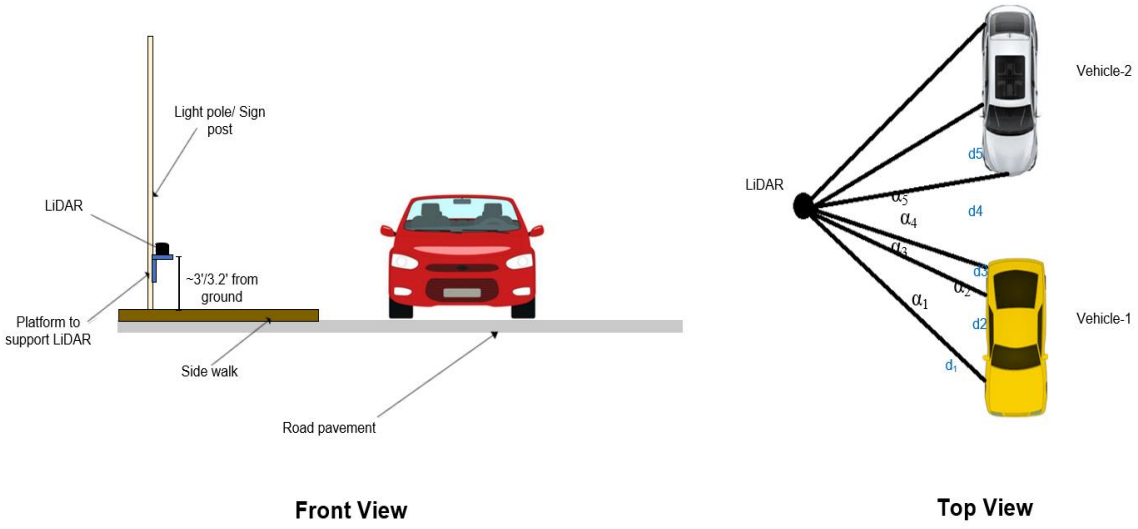


Figure 5.3 LiDAR sensor installation at an intersection for turning movement data collection.

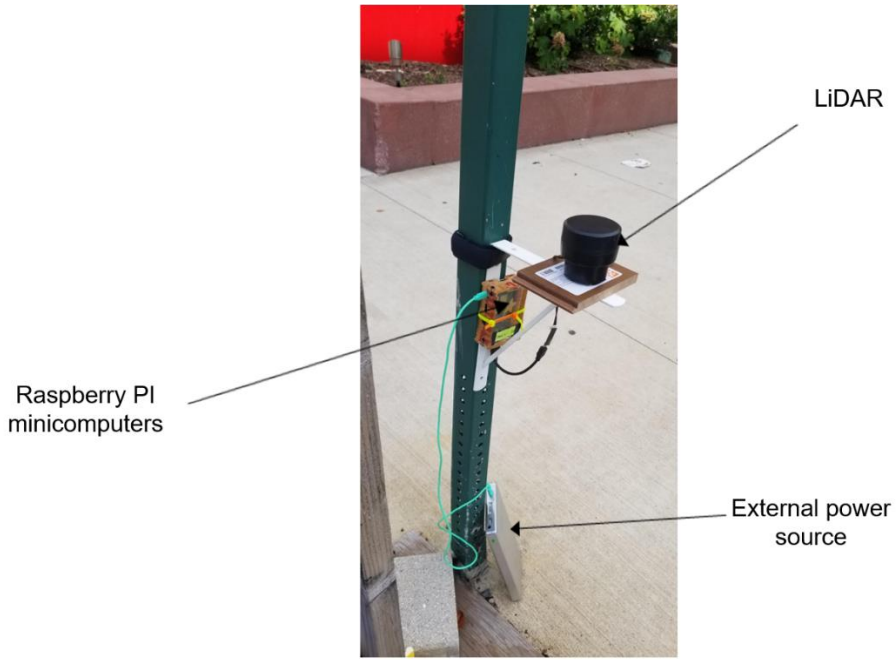


Figure 5.4 A LiDAR sensor placement at the intersection 1.

The LiDAR sensors are installed at each corner of the intersection to cover all the approaches. A total of three and four LiDAR sensors are installed at intersection 1 and intersection 2, respectively, as shown in Figure 5.5 and Figure 5.6. All LiDAR sensors are connected to the raspberry pi minicomputers, which indeed connected to the Wi-Fi hotspot to synchronize the clock of sensors. Synchronization of the timestamp is essential since it allows to identify multiple detections of a discrete vehicle.

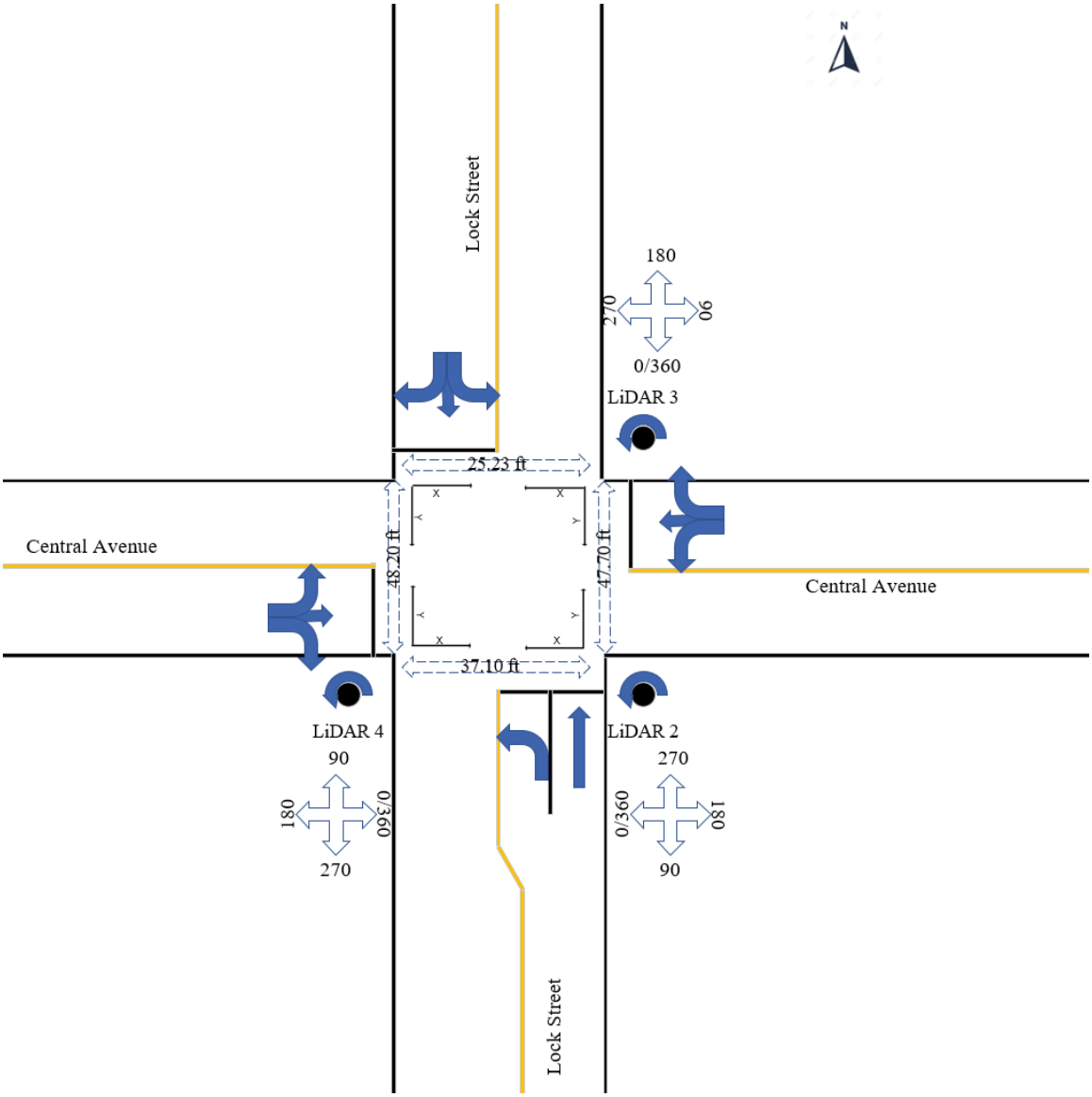


Figure 5.5 Graphical representation of intersection 1 with lidar placements.

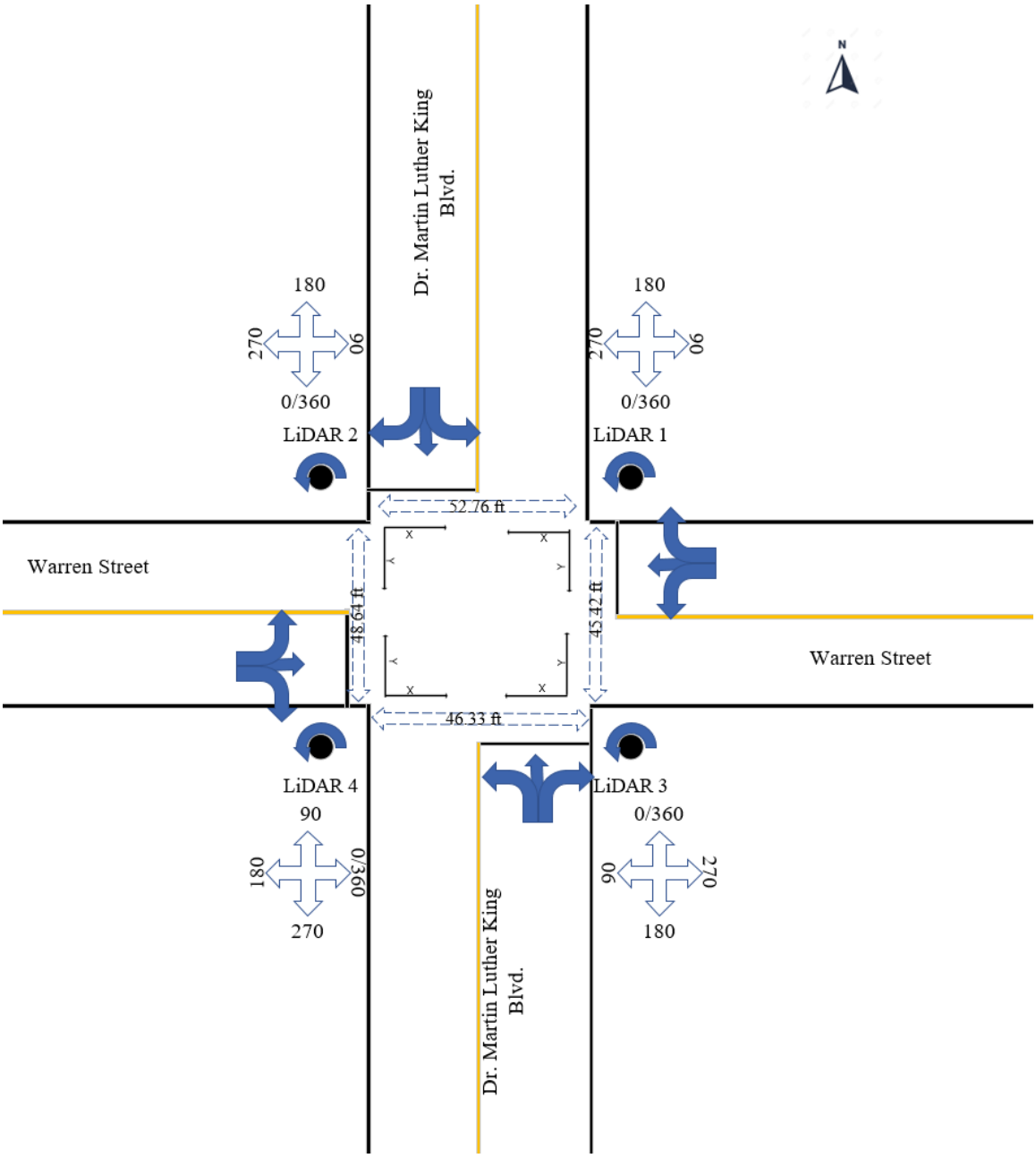


Figure 5.6 Graphical representation of intersection 2 with lidar placements.

CHAPTER 6

RESULTS

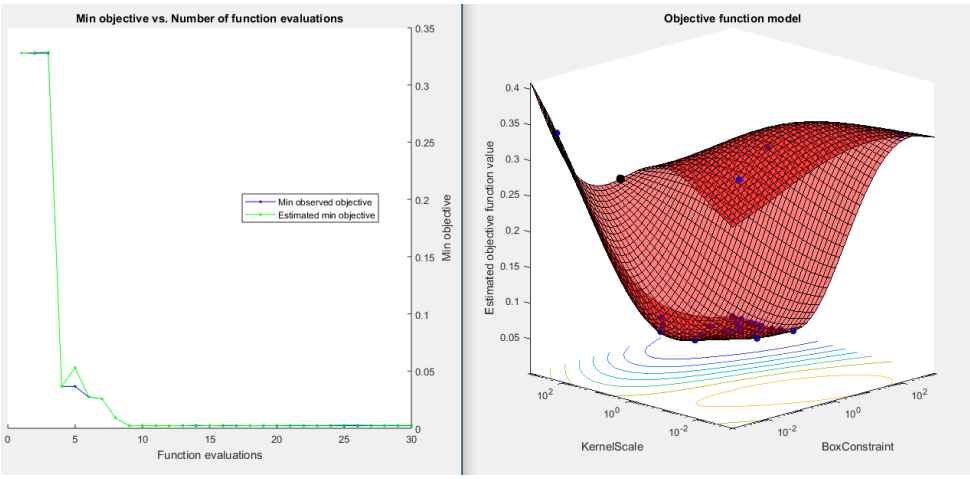
6.1 LiDAR Based Traffic Counts

For the analysis, one second of the time interval is selected, and the corresponding distance profile is transformed into a signal to capture the small changes using Equations (3.9). The mean and maximum energy for each signal is calculated by Equation (3.10). During the POC tests, it is discovered that the vertical curves and the smoothness of roadway pavement are one of the factors affecting capturing the accurate distance from a LiDAR sensor. Due to each location's discrete road condition, the mean energy values vary for individual location even though a LiDAR sensor is placed at the same height. By conducting manual observation, the threshold values (τ_n) are identified to determine each signal as detection and non-detection cases. τ is the threshold value, and n is the corresponding location. The selected threshold values are$\{\tau_1, \tau_2, \tau_3\} = \{120, 50, 4\}$. During the grouping stage, the detection instances are defined as case- {1} and non-detection instances are defined as case- {0}.

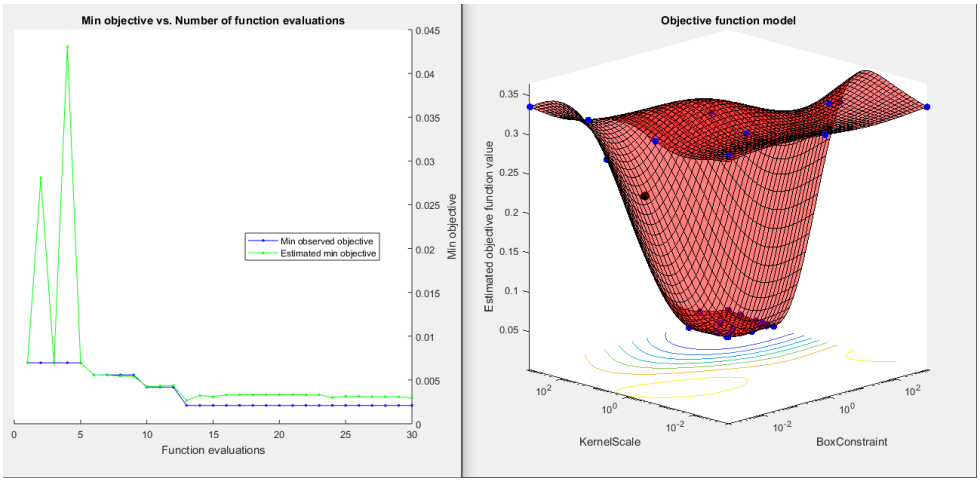
It is noted that the color of a vehicle's exterior surface affects the reflection of light rays towards the LiDAR sensor. That is, the black and metallic silver-colored surfaces are less reflective. Such kind of color property causes missing detections than a white car, at the same distance. However, in this study, such

instances are handled by comparing a minuscule change in the calculated energy with recorded videos.

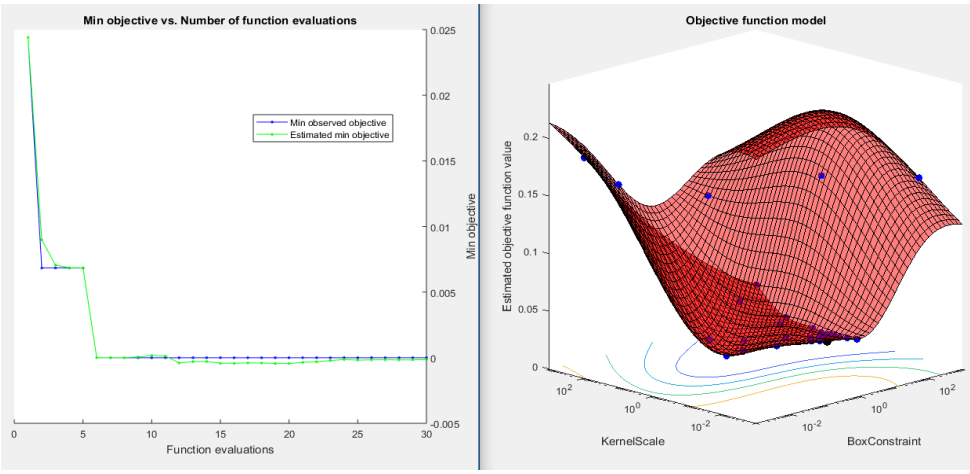
Using the threshold values, 50% of the collected data is used to train the SVM, and the residual 50% of the data is used for testing (predict the case) purpose, resulting in 1403 and 1404 data points for training and testing, respectively. During the SVM training, the primary objective is to increase the width of the plane between two cases of data using Equation (3.13 to 3.15). For each location, the SVM is trained independently. Figure 6.1 shows the obtained optimal results to select the kernel parameter (σ) and soft margin parameter (C).



(a) Objective function results for location1



(b) Objective function results for location 2



(c) Objective function results for location 3

Figure 6.1 SVM results.

Table 6.1 shows the estimated detection instances for each location and its comparison with the ground truth data. The result demonstrates that the proposed method can achieve an accuracy of 83% and higher.

Table 6.1 Results Obtained from Proposed Method and Comparison with Ground Truth

Location	Estimated Detection Instances	Ground Truth Count	Accuracy (%)
1	456	549	83.1%
2	332	352	94.3%
3	157	183	85.8%

Moreover, the accuracy of the developed methodology is compared with the current technologies. Table 6.2 shows that the recommended method can achieve the accuracy close to the modern technologies for vehicle count.

Table 6.2 Traffic Count Accuracy Comparison with Current Technologies

	Proposed Methodology	RTMS [69,70]	Loop Detector* [71]	Video Image Processing [72]
Accuracy (%)	85% - 95%	90%-95%	85.2%-90.6%	73%-87%

*: average error rate of vehicle counting decreases as interval increases (e.g., 20 sec to 15 minutes)

The impact of vehicles' exterior colors on the accuracy of the proposed method is also examined, as summarized in Table 6.3. It is observed that bright colors, such as white and silver, achieved relatively higher accuracy than dark colors (e.g., black and blue).

Table 6.3 Impact of Exterior Colors on Accuracy

Color	Actual (a)	Detected (b)	Accuracy (100*(a/b))
White	188	175	93 %
Silver	190	169	89 %
Black	329	275	83 %
Grey	177	150	85 %
Blue	35	28	79 %
Red	87	75	86 %
Brown	42	35	84 %
Others	35	26	74 %

6.2 LiDAR Based Turning Movement Counts

First, the collected LiDAR data are processed independently from each other to capture the vehicle trajectories using single LiDAR by applying the steps described in section 4.2. The nearest LiDAR for each movement is used to capture the turning movement. For the application of the inverse sensor model to localize the detected vehicles, a grid map of $50\ feet \times 50\ feet$ is defined by defining LiDAR position at (0,0). The cell dimension is defined as $1\ feet \times\ feet$. Figure 6.2 shows the range of the variation along the primary axis, which is used to define the movement of the vehicles at the intersection. The obtained turning movement counts from each intersection is then compared with ground truth count, which is obtained using video recording.

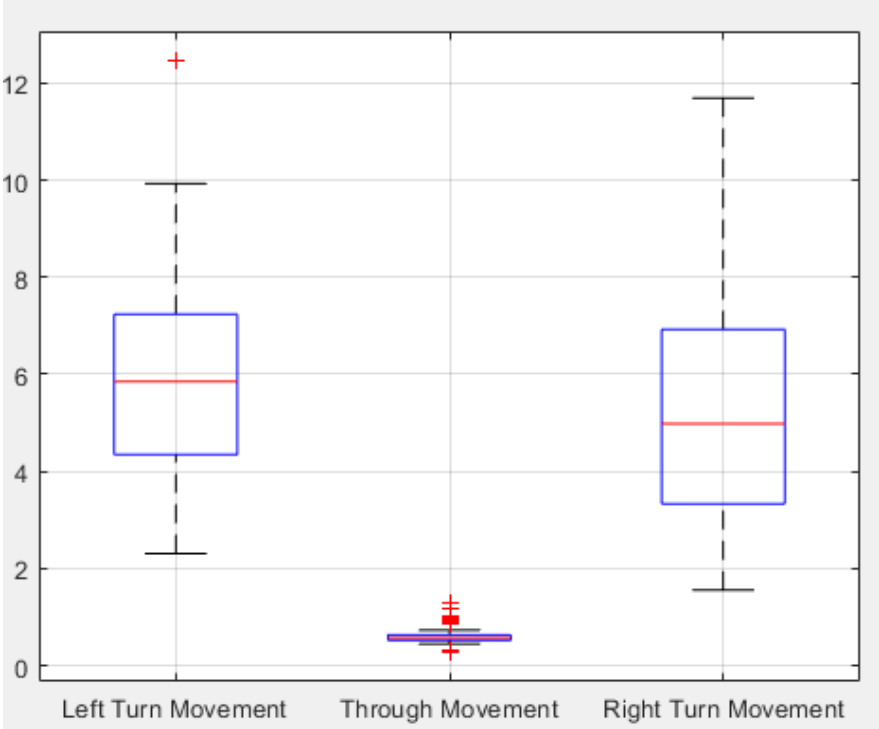


Figure 6.2 Observed range of variations along primary axis at each intersection.

It is noticed that the distribution of the variations along the primary axis has normal distribution, that justifies the application of discrete Kalman Filter with the assumption of gaussian distribution of data. Table 6.4 shows the accuracy of turning movement count for each direction achieved using its nearest LiDAR for intersection - 1. The nearest LiDAR is defined as the immediate LiDAR for an entrance approach to an intersection. During the data processing, it is observed that the accuracy for all the left-turning movements is below 75%. Movements within the intersection are studied to improve the accuracy of the proposed method, which is not captured by the nearest LiDAR due to occlusion or missing data points but captured in the middle of the intersection or at the exit of an intersection by another LiDAR. A timestamp for a detected vehicle is used to

reduce the double count errors since each LiDAR sensor is synchronized with the Wi-Fi clock. Table 6.5 shows the accuracy for each turning movement at the intersection – 1 after considering all LiDAR data.

Table 6.4 Turning Movements Count Accuracy Based on Nearest LiDAR/ Single LiDAR at Intersection - 1

		Accuracy (%)		
Direction	**Nearest LiDAR**	**Left**	**Through**	**Right**
NB	2	69.44%	82.32%	80.00%
EB	4	55.55%	86.50%	82.60%
WB	3	57.89%	85.71%	77.77%
Overall accuracy			75.31%	

As shown in Table 6.4, the overall accuracy of the methodology at an intersection – 1 is improved from 75.31% to 87.31%. Moreover, it is noticed that the accuracy of all left-turning movements is improved significantly. Figure 6.3 to Figure 6.5 shows the comparison between ground truth obtained data and turning movement count obtained from the proposed methodology.

Table 6.5 Turning Movements Count Accuracy using all LiDAR at Intersection –1

	Accuracy (%)		
Direction	**Left**	**Through**	**Right**
NB	83.33%	88.23%	80.00%
EB	77.77%	89.68%	91.30%
WB	72.22%	87.50%	77.77%
Overall accuracy		87.15%	

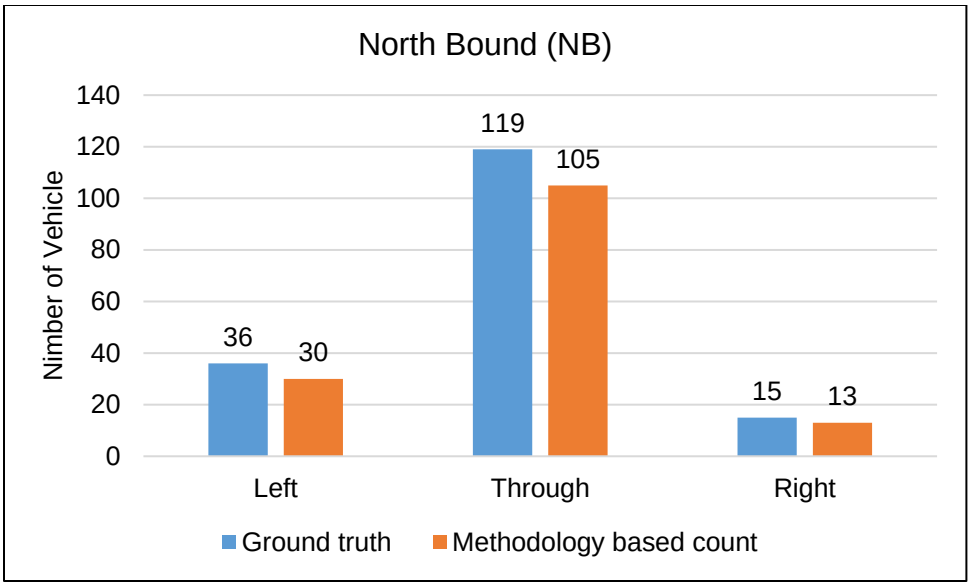


Figure 6.3 Comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 1) for NB.

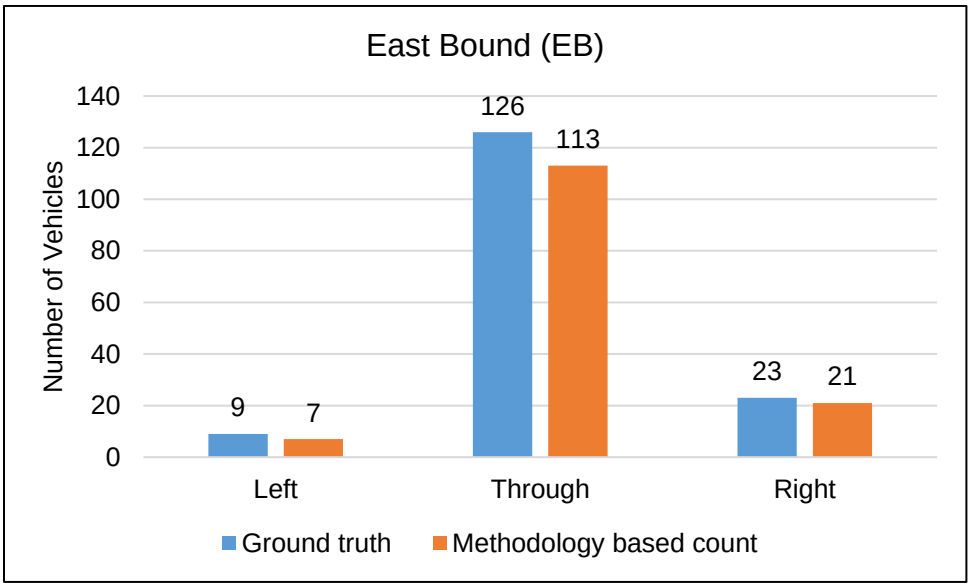


Figure 6.4 Comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 1) for EB.

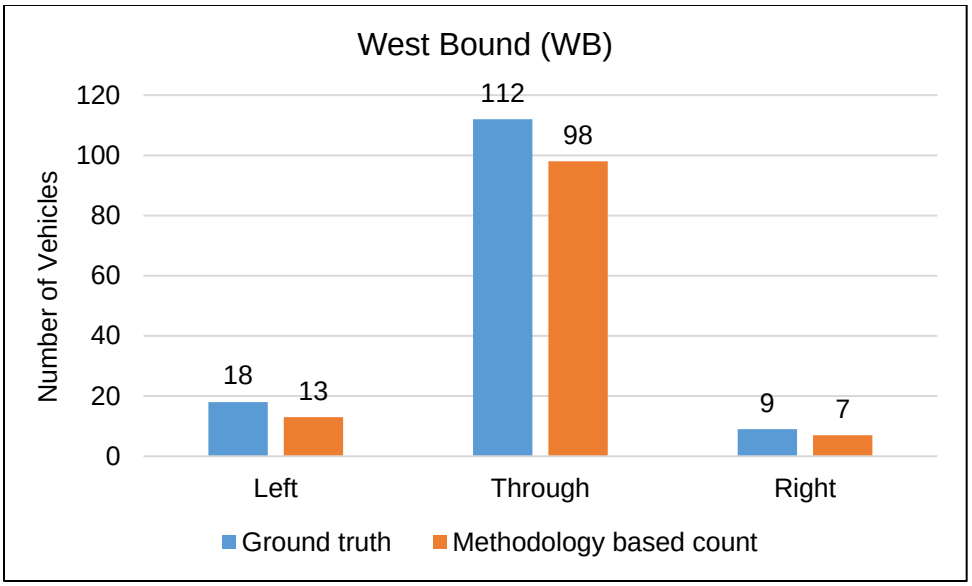


Figure 6.5 Comparison between ground truth turning counts vs. proposed methodology based turning movement counts (intersection – 1) for WB.

Table 6.6 and 6.7 shows the obtained accuracy of the proposed methodology at an intersection – 2 obtained from the nearest LiDAR and considering all LiDAR data. Unlike the intersection – 1, the intersection – 2 has more observed pedestrian activity. Furthermore, the intersection – 2 is wider compare to the intersection -1, and low accuracy has been noticed compare to an intersection – 1. Figure 6.6 to Figure 6.9 shows the comparison between ground truth obtained data and turning movement count obtained from the proposed methodology at an intersection – 2.

Table 6.6 Turning Movements Count Accuracy Based on Nearest LiDAR/ Individual LiDAR at Intersection - 2

		Accuracy (%)		
Direction	**Nearest LiDAR**	**Left**	**Through**	**Right**
NB	3	61.23%	72.56%	75.56%
EB	4	54.78%	81.76%	74.39%
WB	1	55.23%	79.89%	81.15%
SB	2	59.65%	72.56%	74.12%
Overall accuracy			70.15%	

Table 6.7 Turning Movements Count Accuracy using all LiDAR at Intersection –2

	Accuracy (%)		
Direction	**Left**	**Through**	**Right**
NB	80.00%	79.70%	77.41%
EB	78.26%	85.72%	76.92%
WB	77.77%	83.33%	86.36%
SB	75.55%	78.85%	77.35%
Overall accuracy		79.76%	

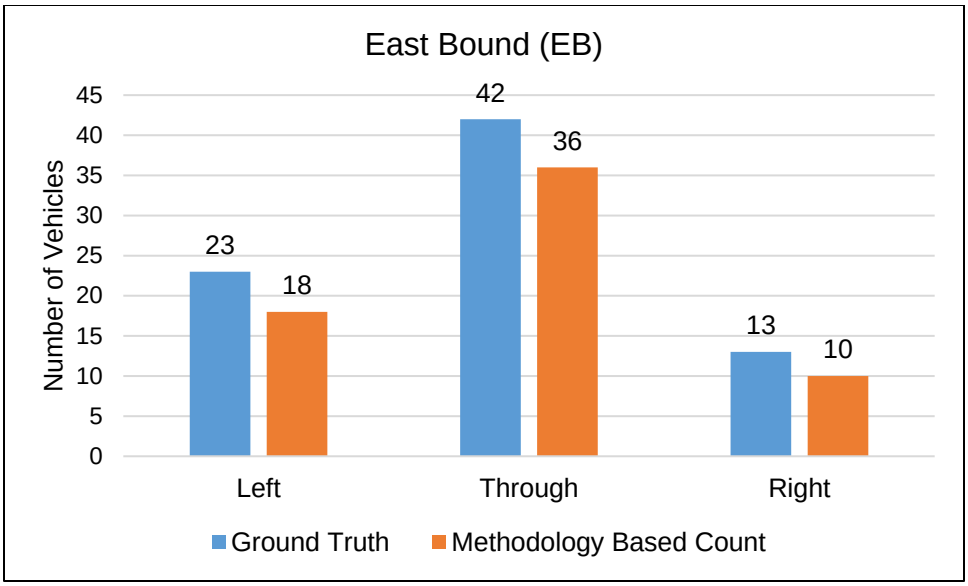


Figure 6.6 Comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 2) for EB.

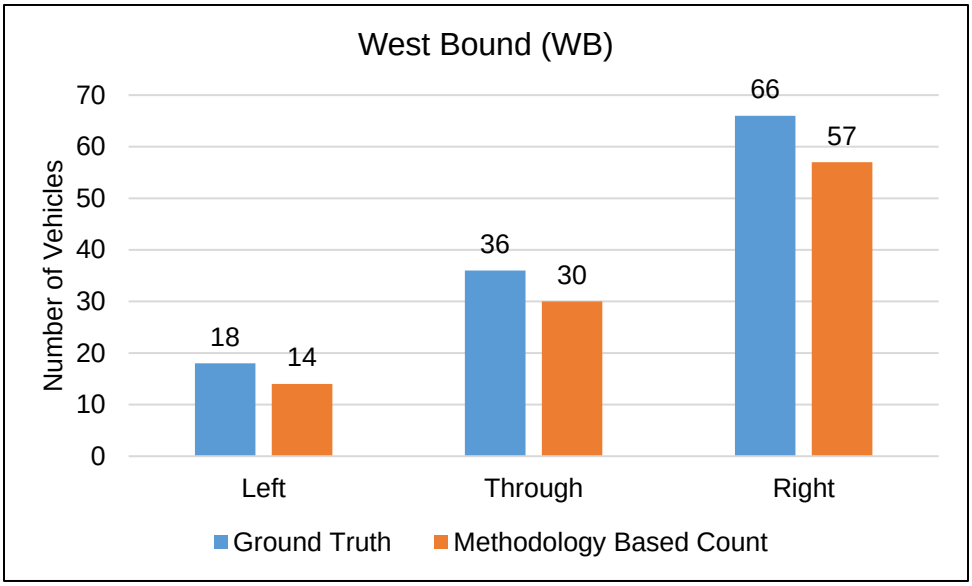


Figure 6.7 Comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 2) for WB.

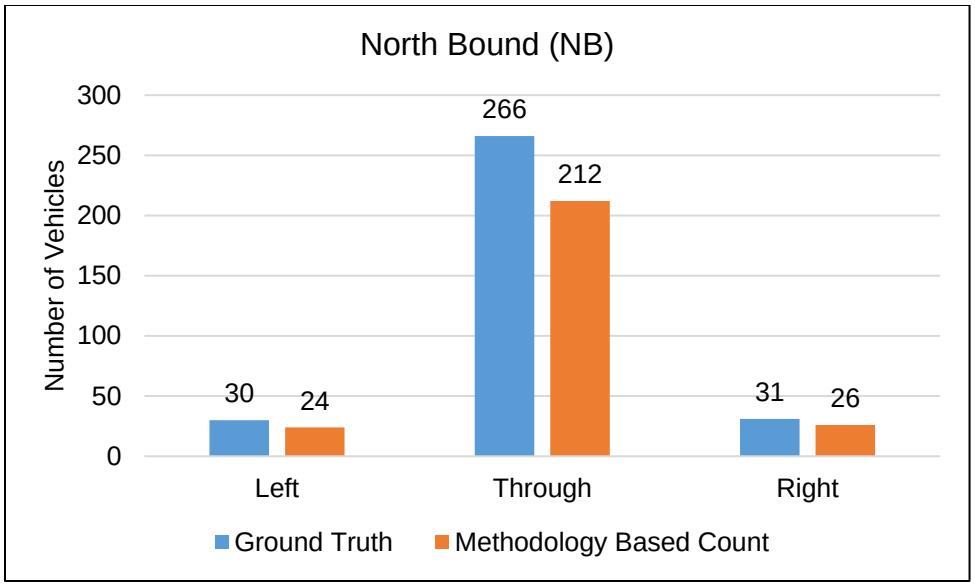


Figure 6.8 comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 2) for NB.

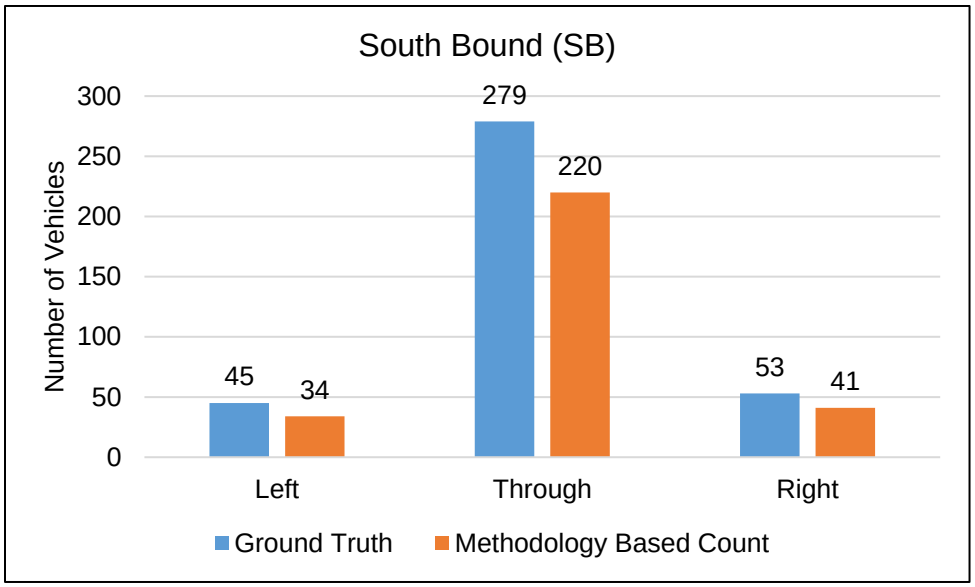


Figure 6.9 Comparison between ground truth turning counts vs. proposed methodology based turning movements count (intersection – 2) for SB.

Figure 6.10 and Figure 6.11 represents the R-squared value for the X and Y axis between proposed method-based localization points and reference points. The data are selected randomly from both the location to study the relation

between LiDAR-based reference points of each axis vs. model-based localized points. The obtained R-squared value ranges between 0.87 and 0.89, which shows that the developed methodology can localize the detected object accurately.

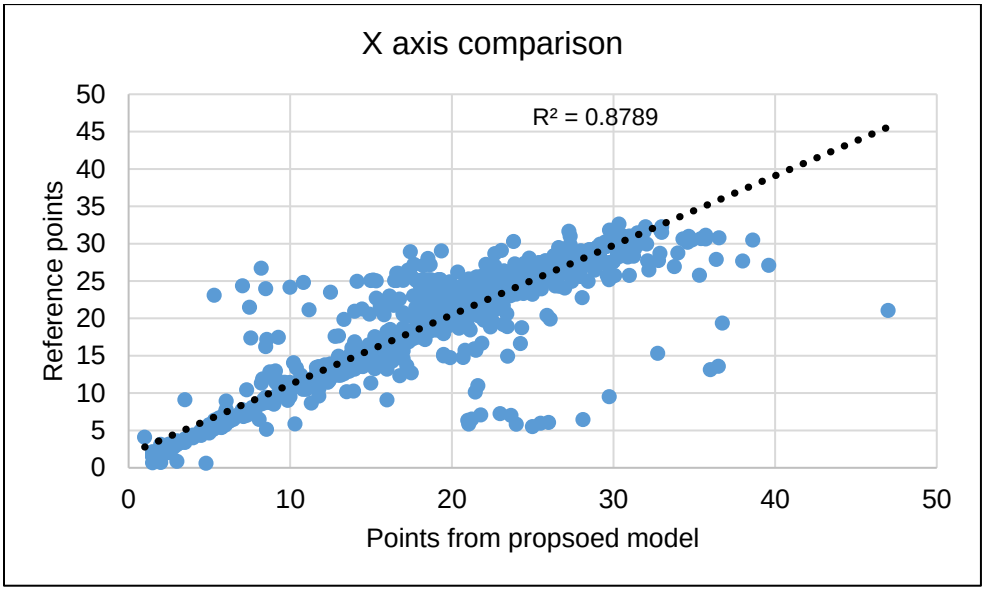


Figure 6.10 Comparison between LiDAR based reference points of X axis vs. proposed model based localized points.

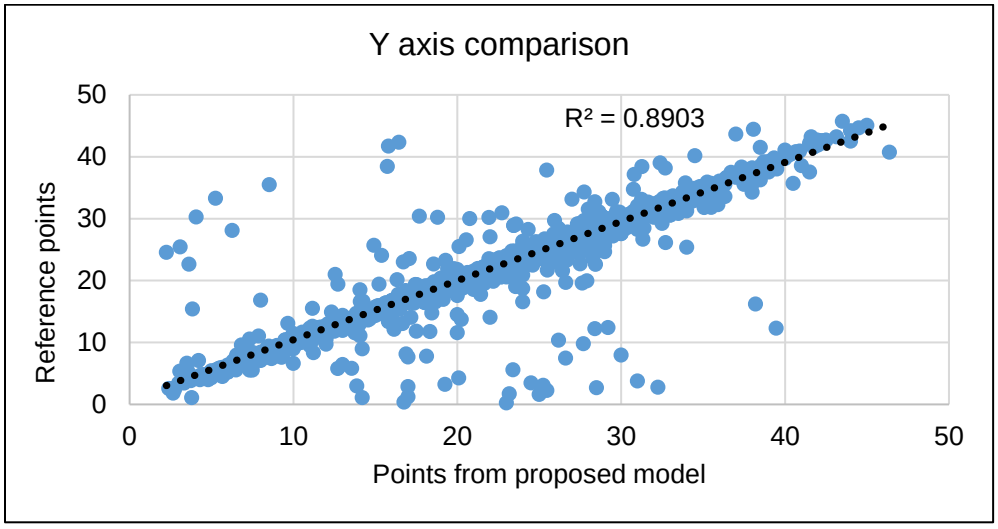


Figure 6.11 Comparison between LiDAR based reference points of Y axis vs. proposed model based localized points.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

This chapter comprises the in-depth discussion of conducted research work and future direction. Moreover, a primary cost comparison is also conducted to illustrate the cost effectiveness of the proposed approach.

7.1 Conclusions

Two methodologies are proposed to utilize a low-cost 2-D LiDAR sensor for traffic count and turning movement count data collection at an intersection. An approach using a support vector machine (SVM) and continuous wavelet transform (CWT) is proposed for traffic count. The data from three intersections located in Newark, New Jersey, are used to examine the efficiency of the proposed traffic count methodology. A single direction data is captured from the intersections, as mentioned in Table 5.1. The LiDAR is installed at the height of thirteen feet from the ground, which scans a vertical plane and is powered by a laptop. A total of 142 minutes of data are collected. The proposed methodology is applied to the collected data and predicted traffic count is compared with ground truth. The comparison shows that the proposed methodology can achieve accuracy between 83% ~ 95%. However, it is noticed that the accuracy of the methodology is affected by the exterior color of the vehicles. The conducted study shows that the white-colored vehicles are accurately detected in the methodology with an accuracy of 93%; on the other side, blue-colored vehicles have the least detection with 79%

accuracy. The accuracy of the proposed methodology is compared with the current technologies such as RTMS, Loop detector, and Video Image processing. The comparison illustrates that the developed method can achieve acceptable accuracy with 2–D LiDAR sensor.

The K-Means clustering method, inverse model, and Kalman filter-based methodology are proposed to detect, localize, and track a vehicle movement within the intersection. The data from two intersections are collected to study the effectiveness of the proposed turning movement count methodology. Unlike the LiDAR installation for traffic count data collection, the LiDAR is placed to scan the horizontal plane for this data collection activity. The Lidar sensors are mounted at the height of 3' ~ 3.2'. A maximum of three LiDAR sensors is used at an intersection during the data collection—the LiDAR sensors placed at each corner of an intersection. The sensors are connected to the raspberry pi minicomputer to store the data. The raspberry pi is connected to the external power supply, and a Wi-Fi network to accurately synchronize the system clock. The conducted data analysis shows that the proposed methodology can accurately detect, localize, and track 79% ~ 88% of vehicles within the intersection. First, the movement counts obtained from individual LiDAR are compared with ground truth. The comparison indicates the lower accuracy (<75%) for the left-turning movement for each direction, mainly caused due to the occlusion and missing data. All LiDAR data are considered to improve the accuracy. The after comparison demonstrates the improved accuracy of the left-turning movement, ranging between 72% ~ 82%.

The primary objective of this thesis is to develop a cost-effective data collection methodology using low-cost sensors. A cursory cost comparison is conducted, as shown in Table 7.1, to study the cost-effectiveness of the proposed method. The cost of the used LiDAR sensor array incurred the cost of a Scanse sweep LiDAR sensor, Raspberry pi mini-computer, external power source, and additional mounting gear. A Scanse sweep LiDAR is available at the cost of $ 250 [73]. The cost of raspberry pi varies from $ 40 to $ 70 [74]. The external power source costs $ 200 [75], and additional mounting gear cost is considered as $ 150. The cost comparison illustrates that the proposed methodology can achieve acceptable accuracy at low cost which allows the planning authorities to conduct the data collection at large scale with low investments.

Table 7.1 Accuracy and Cost Comparison of Proposed Methodology vs. Current Technologies

	Proposed Methodology	Miovision* [76,77,78]	Manual Count [76,77]	Pneumatic tube counter* [76,77]	RTMS* [76,77]
Cost/ Unit or Cost/ Person ($)	670 ~ 750	3,000 ~ 4,000	500 ~ 1,000	500 ~ 700	+ 5,000
Accuracy (%) for traffic count	83% ~ 94%	93% ~ 95%	97% ~ 99%	93% ~ 95%	90% ~ 95%
Accuracy (%) for turning movement	79% ~ 88%	93% ~ 95%	97% ~ 99%	NA	NA

*The cost includes only a unit price; it does not incur additional expenses such as maintenance, power supply, installation crew charges.

Furthermore, the accuracy of the proposed method is compared with the recent research works that used 16, 32, or 64 channels 3 – D LiDAR to conduct traffic data collection, as shown in Table 7.2. As described earlier, the primary purpose of the conducted research is to develop the methodologies to handle low density 2–D LiDAR points cloud for traffic data collection, a cost comparison is made between 2–D LiDAR sensors and 3–D LiDAR sensors that are available in the market. It is observed that the existing 2–D LiDAR sensors available in the market costs between $ 400 to $ 2,000 per unit [83], whereas the cost of a single unit of 3–D LiDAR sensors exceeds $ 7,000 [83, 84].

Table 7.2 Accuracy Comparison of Proposed Methodology with Recent Research Works

	Proposed Methodology	Xu et al. [79]	Yang [80]	Bandaru [81]	Sualeh et al. [82]
Accuracy (%) for detection	83% ~ 94%	95% ~ 97%	93%	90 %	88%
Accuracy (%) for tracking	79% ~ 87%	94% ~ 97%	NA	96% ~ 98%	88%

7.2 Future Research

From the conducted data collection and data analysis following recommendations are made for future research:

1) Traffic count data:

   - Investigate the signal strength data retrieved along with the distance information from the LiDAR sensor for further analysis.
   - Focus on the development of the proposed methodology more robust, which will be not location specific.

- Increase the data collection period for extended time interval with various locations, such as at the mid-way of freeway, at entry or exit ramp.

- A large training data sets of detection and non-detection cases should be used to train the Support Vector Machine (SVM) that can improve the accuracy of proposed method.

- Consider various weather conditions (e.g., rainy, cloudy) and daytimes (e.g., morning, evening, or night) to evaluate the impact of different conditions on proposed methodology.

2) Turning movement count:

- A constant velocity model based Kalman filter is used in the proposed approach, for the future study an unscented Kalman filter should be consider.

- Different types of signalized intersections with various lane configuration should be consider.

- Increase the number of LiDAR sensors at an intersection to study the impact on accuracy.

Considering above recommendations, a brief cost-benefit analysis should also be conducted, that includes the data collection duration.

APPENDIX A

PSEUDO CODES FOR VEHICLE TRAJECTORY CONSTRUCTION

This section provides in detail pseudo code of Inverse Sensor Model, K-Means Clustering, and Kalman Filter.

1. Following is the pseudocode for the Inverse Sensor Model (State model)

Algorithm for Inverse Sensor Model $(x_i, y_i, t_i, \theta_i, d_i)$

known LiDAR location (x_1, y_1)
set the cell dimensions (r) on a grid map $r = 1$
for each elapsed time t_i LiDAR obtain (θ_i, d_i)
if θ_i in field of view and d_i >= 1 and <= 50

$$(x_i)_{t_i} = (\cos\theta_i * d_i)_{t_i} + x_1$$
$$(y_i)_{t_i} = (-\sin\theta_i * d_i)_{t_i} + y_1$$
$$Final\_(x_i)_{t_i} = ceiling((x_i)_{t_i} / r)$$
$$Final\_(y_i)_{t_i} = ceiling((y_i)_{t_i} / r)$$
$$return = (Final\_(x_i), Final\_(y_i))_{t_i}$$

Else
consider t_{i+1}
End
End

2. Following is the pseudocode for K-Means Clustering

Algorithm for K-Means Clustering

Define the number of clusters $k = 2$

for $[x_i, y_i]_{t_i}$

randomly select k centroids: $\mu_1, \mu_2, \ldots\ldots\ldots, \mu_k$

assign each (x_i, y_i) to its closest cluster centroid μ_k

calculate the distance between assign point and cluster:

$c_k = \{n : k = \arg\min_k || x_i - \mu_k ||^2\}$

recalculate the new cluster centers μ_k

$$\mu_k = \frac{1}{c_k} \sum_{i \in c_k} (x_i, y_i)_{t_i}$$

until convergence

End

Return μ_k

3. Following is the pseudocode for Kalman Filter (EKF)

Algorithm for Kalman Filter $(x_i, y_i, v_i)_{t_i}$

for $[x_i, y_i]_{t_i}$

control variable (u_k): $[v_i, v_i]_{t_i}$

initial state: $[x_i, y_i, v_i]_{t_i}$

previous state: $[x_i, y_i, v_i]_{t_i - 1}$

new predicted state: $X_t = Ax_{t-1} + Bu_t + \omega_k$

$$P_{t_p} = Ax_{t-1}A^T + Q_t$$

calculate Kalman gain: $K_t = \frac{P_t H^T}{HP_t H^T + R}$

new observation: $Y_t = CY_{tm} + Z_t$

current state: $[x_i, y_i]_t = X_{t_p} + K_t[Y_t - H * X_{t_p}]$

update the covariance matrix: $P_t = [I - K_t * H] * P_{t_p}$

return: $(x_i, y_i)_{t_i}$

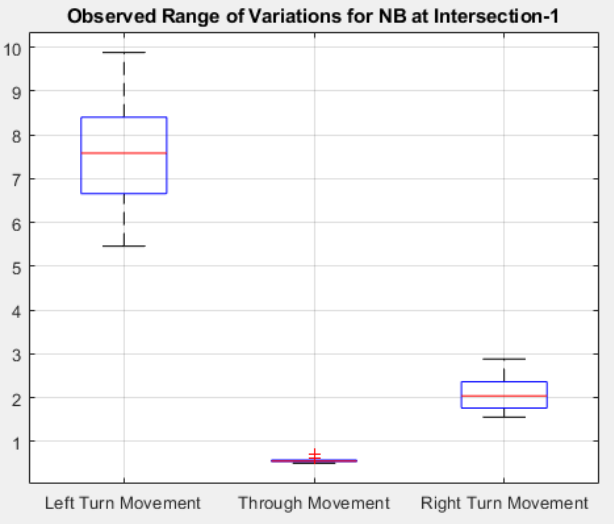
consider current information as a previous location

Repeat

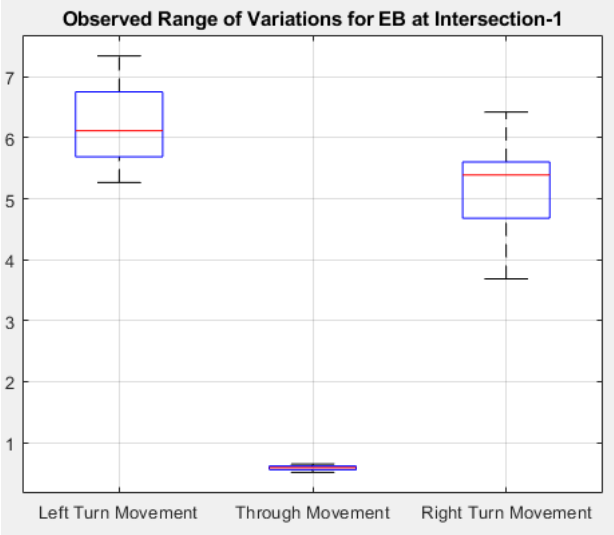
APPENDIX B

RANGE OF OBSERVED VARIATIONS FOR EACH DIRECTION AT INTERSECTION – 1

This section provides observed variations along primary axis for each movement at Intersection-1.

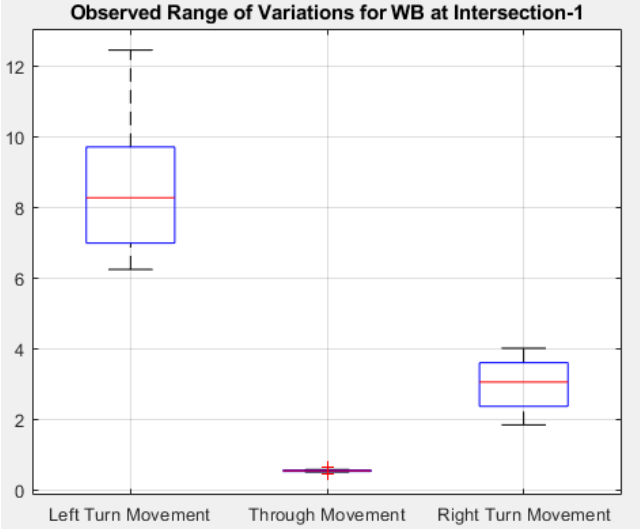


(a) North Bound (NB) Direction



(b) East Bound (EB) Direction

Figure B.1 Range of observed variations at intersection – 1. (Continued)



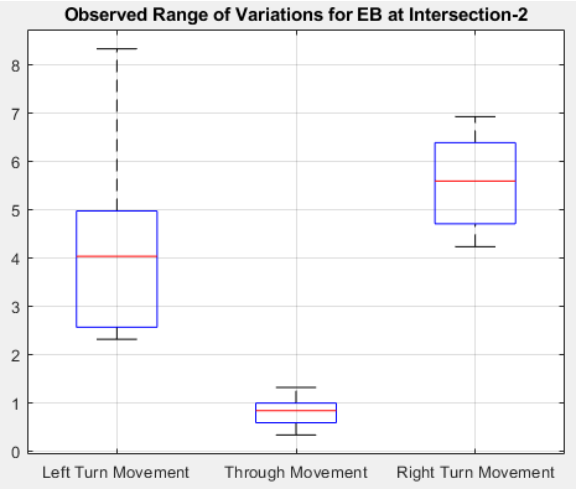
(c) West Bound (WB) Direction

Figure B.1 (Continued) Range of observed variation at intersection – 1.

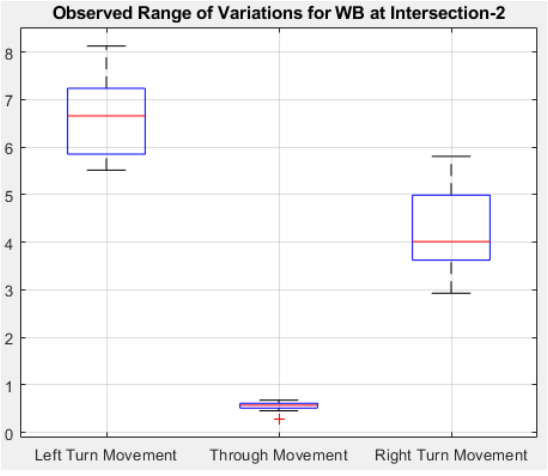
APPENDIX C

RANGE OF OBSERVED VARIATIONS FOR EACH DIRECTION AT INTERSECTION – 2

This section provides observed variations along primary axis for each movement at Intersection-2.

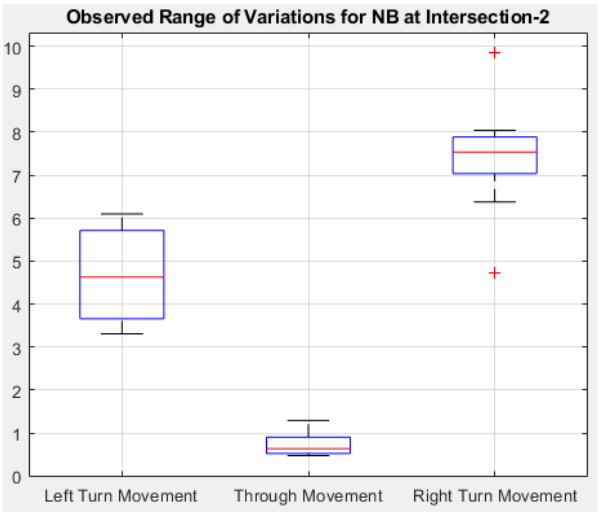


(a) East Bound (EB) Direction

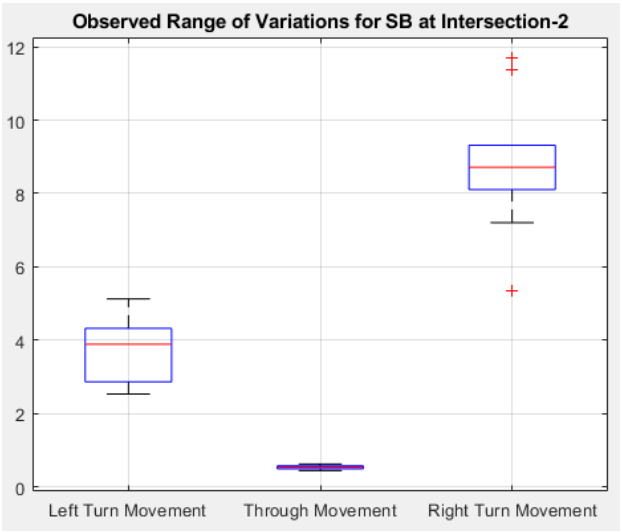


(b) West Bound (WB) Direction

Figure C.1 Range of observed variations at intersection – 2. (Continued)



(c) North Bound (NB) Direction



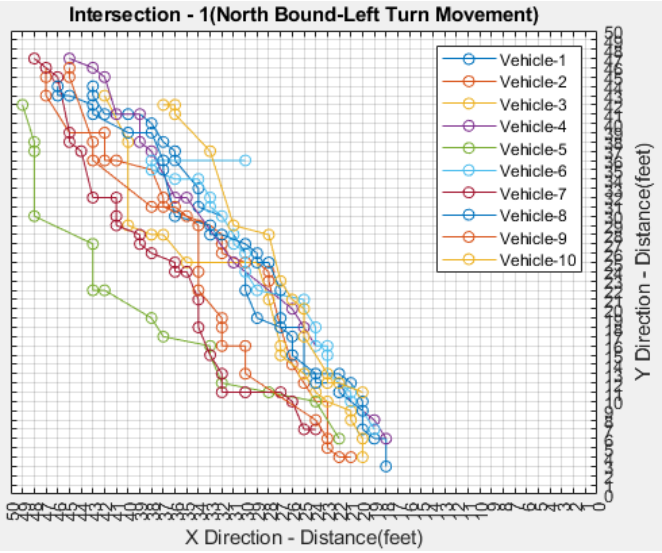
(d) South Bound (SB) Direction

Figure C.1 (Continued) Range of observed variations at intersection – 2.

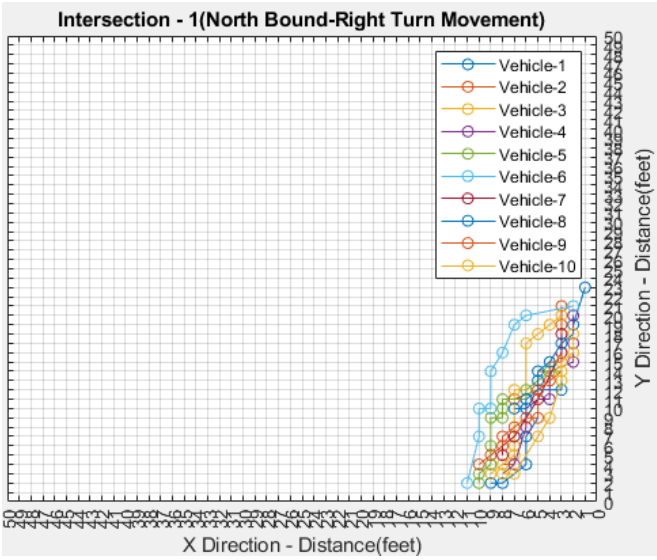
APPENDIX D

SAMPLE TRAJECTORIES FROM INTERSECTION – 1

This section provides a sample trajectory for each movement at Intersection-1.

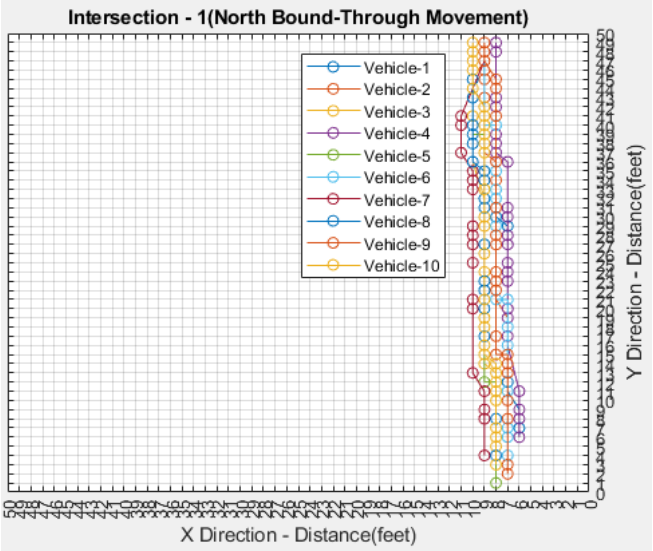


(a) North Bound (NB) Left Turn

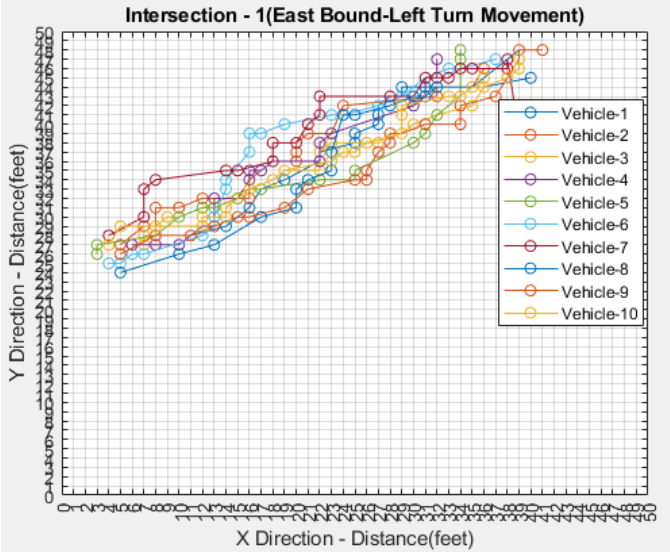


(b) North Bound (NB) Right Turn

Figure D.1 Sample Trajectories from Intersection – 1. (Continued)

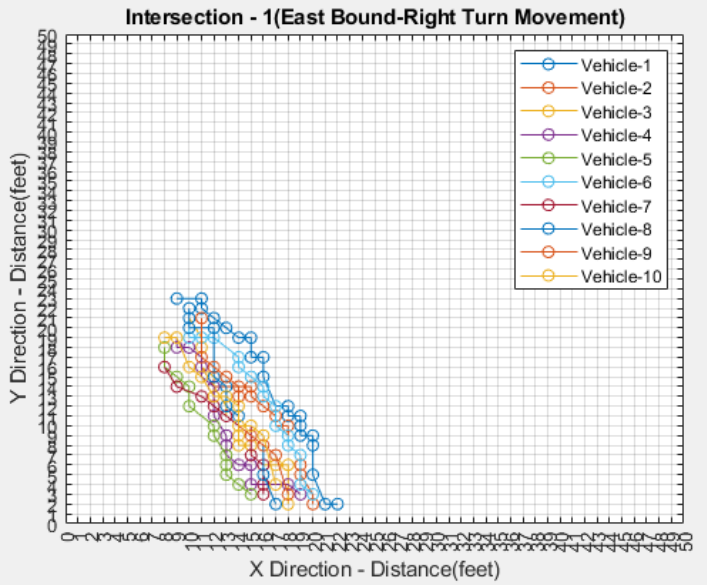


(c) North Bound (NB) Through

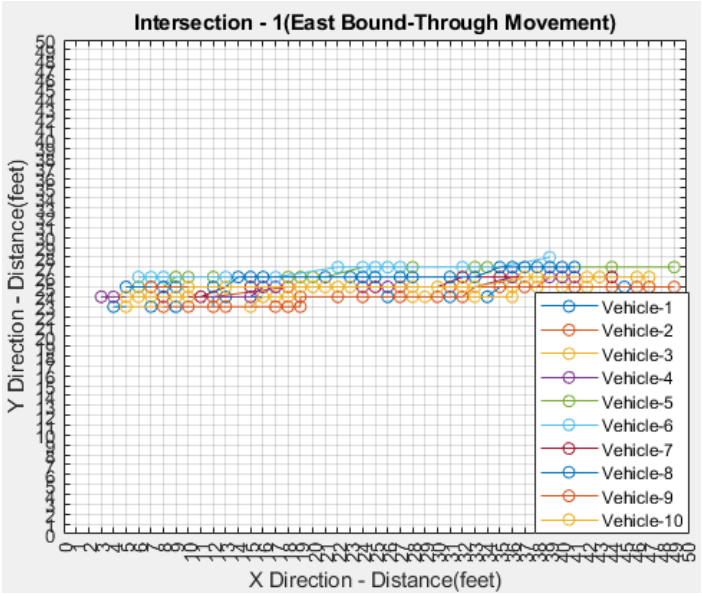


(e) East Bound (EB) Left Turn

Figure D.1 (Continued) Sample Trajectories from Intersection – 1.

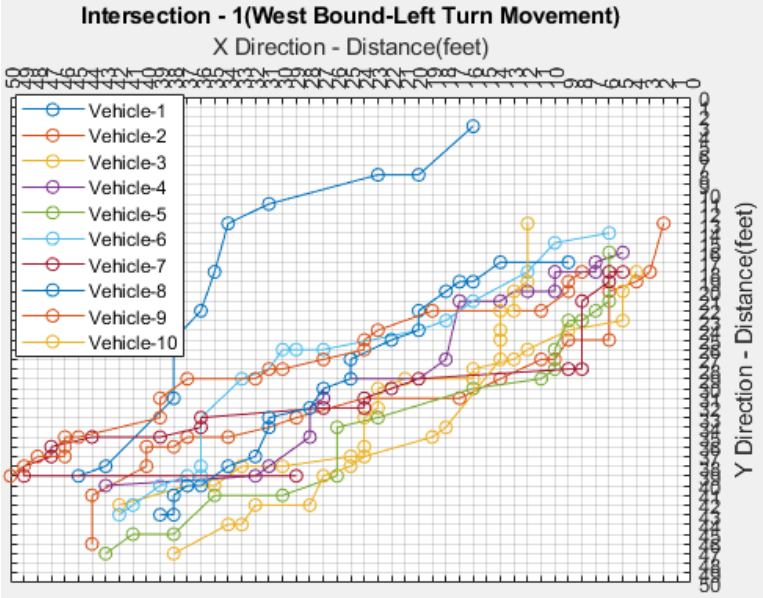


(f) East Bound (EB) Right Turn

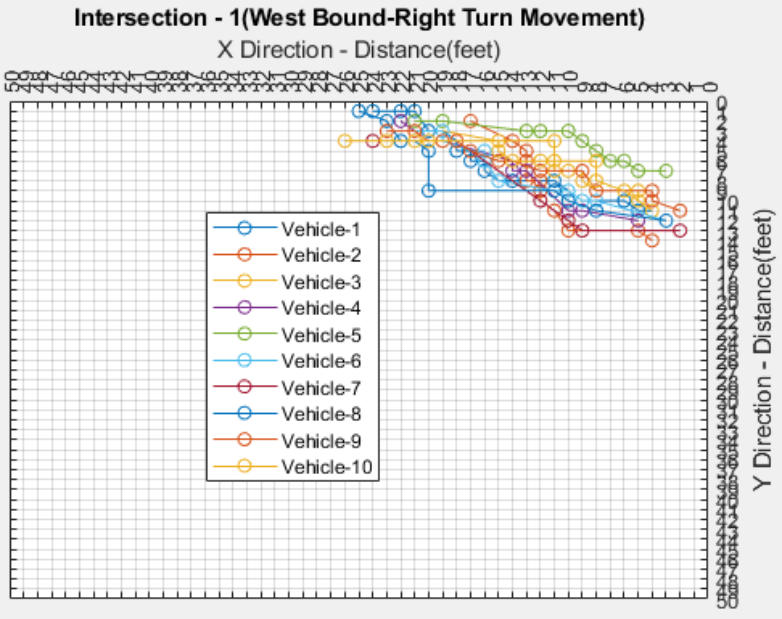


(g) East Bound (EB) Through

Figure D.1 (Continued) Sample Trajectories from Intersection – 1.

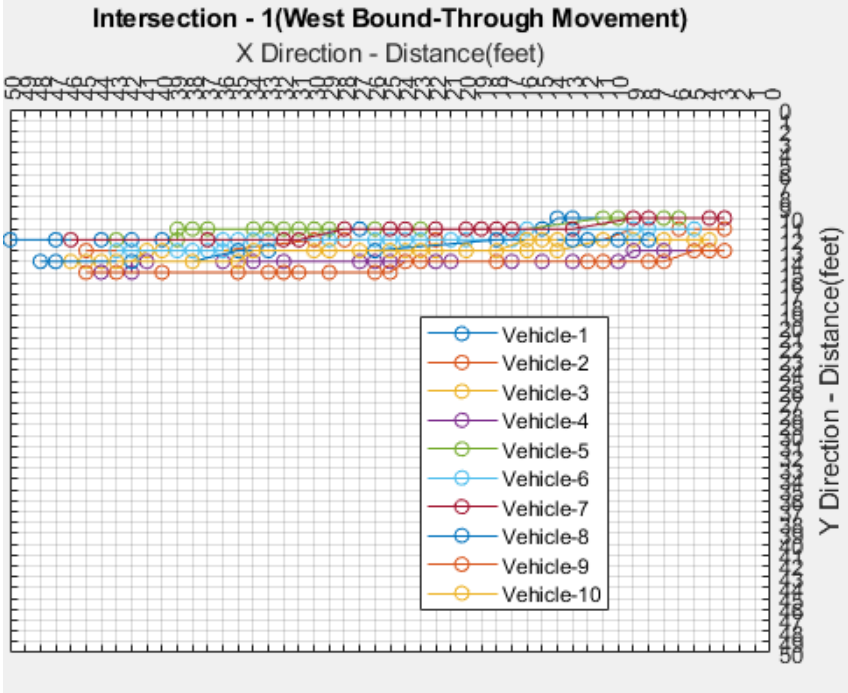


(h) West Bound (WB) Left Turn



(i) West Bound (WB) Right Turn

Figure D.1 (Continued) Sample Trajectories from Intersection – 1.



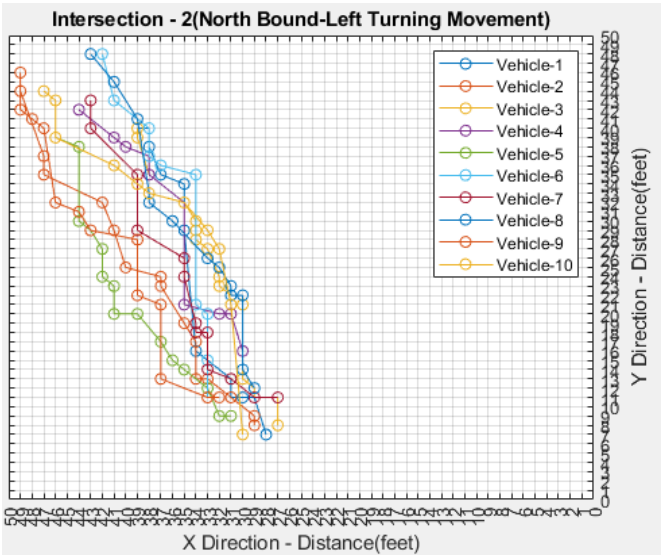
(j) West Bound (WB) Through

Figure D.1 (Continued) Sample Trajectories from Intersection – 1.

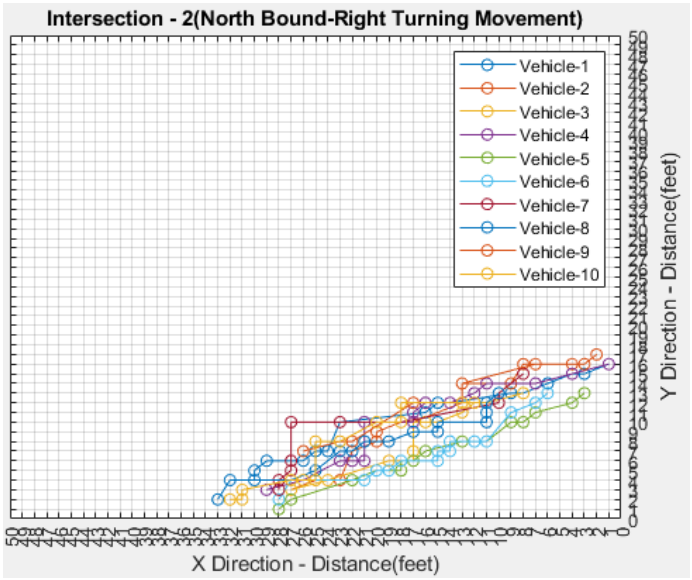
APPENDIX E

SAMPLE TRAJECTORIES FROM INTERSECTION – 2

This section provides a sample trajectory for each movement at Intersection-2.

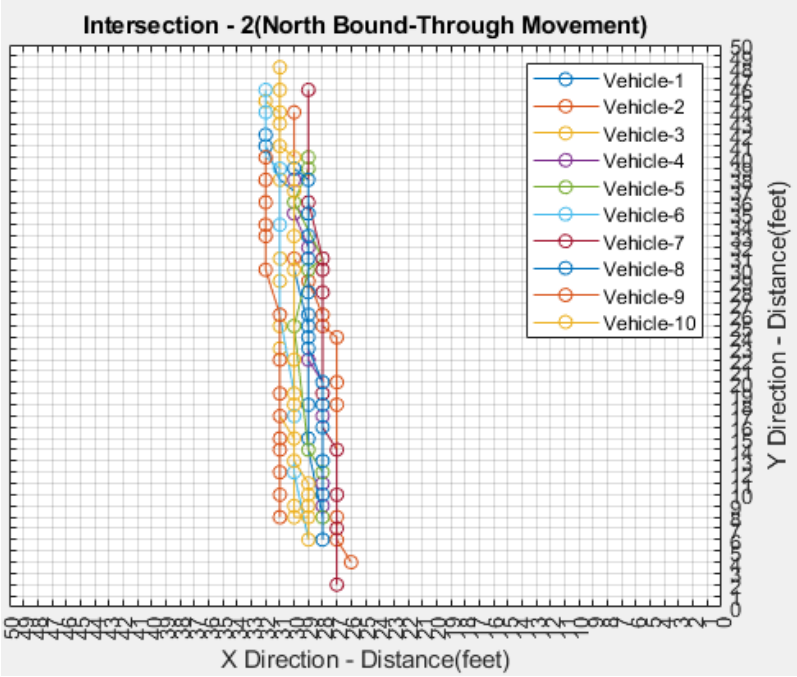


(a) North Bound (NB) Left Turn

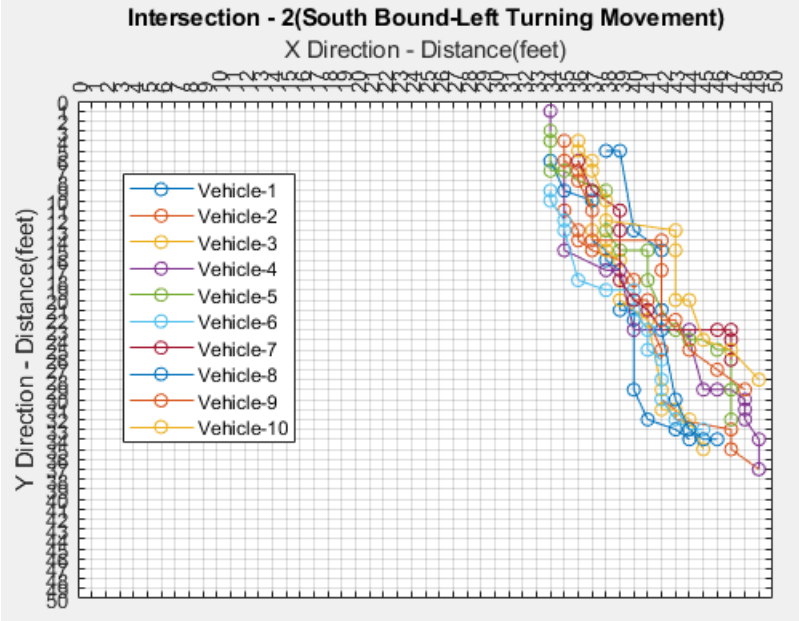


(b) North Bound (NB) Right Turn

Figure E.1 Sample Trajectories from Intersection – 2. (Continued)

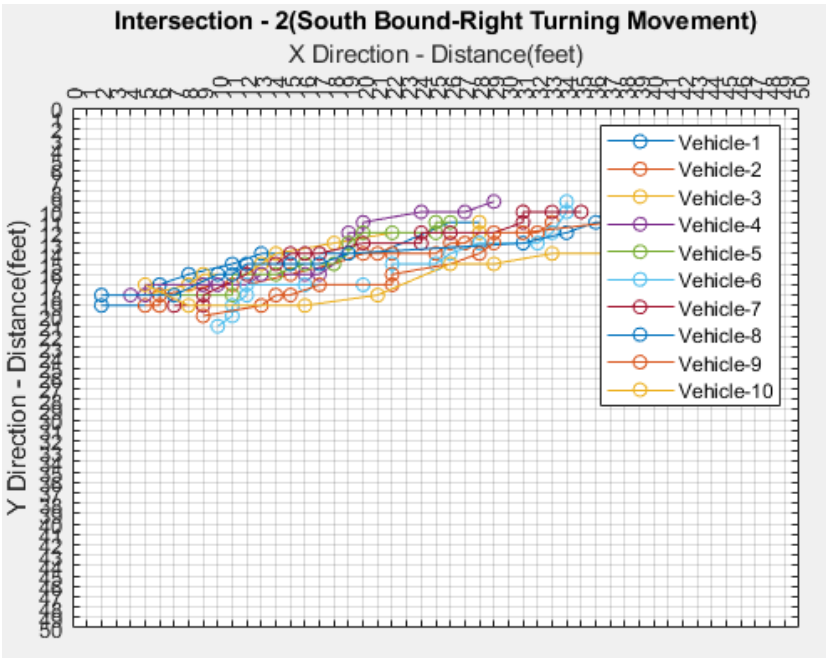


(c) North Bound (NB) Through

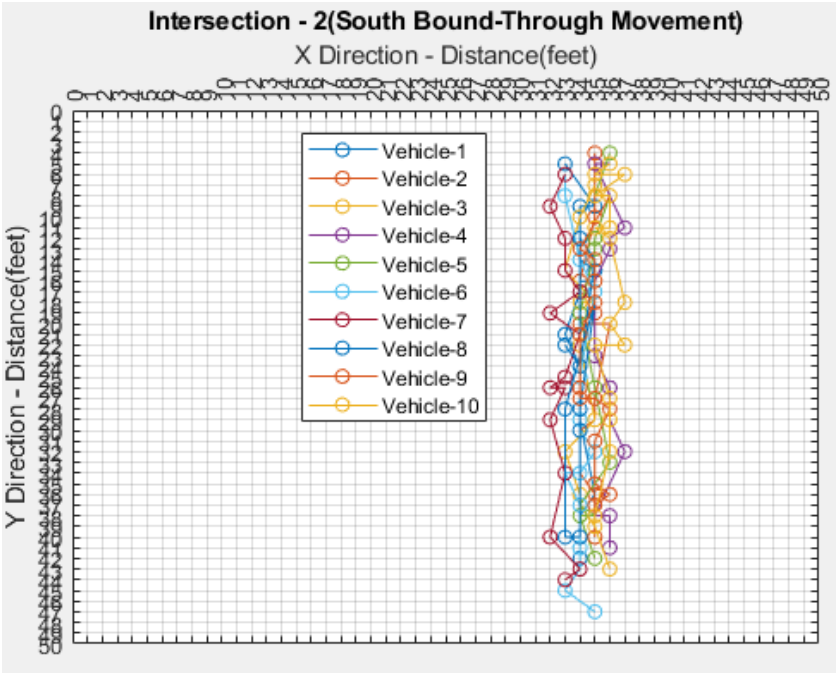


(d) South Bound (SB) Left Turn

Figure E.1 (Continued) Sample Trajectories from Intersection – 2.

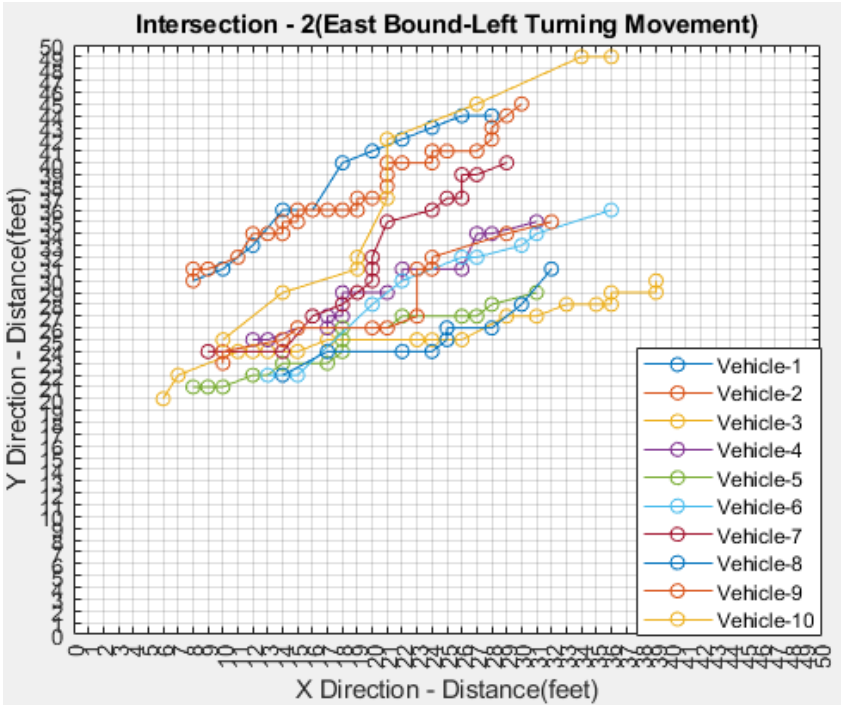


(e) South Bound (SB) Right Turn

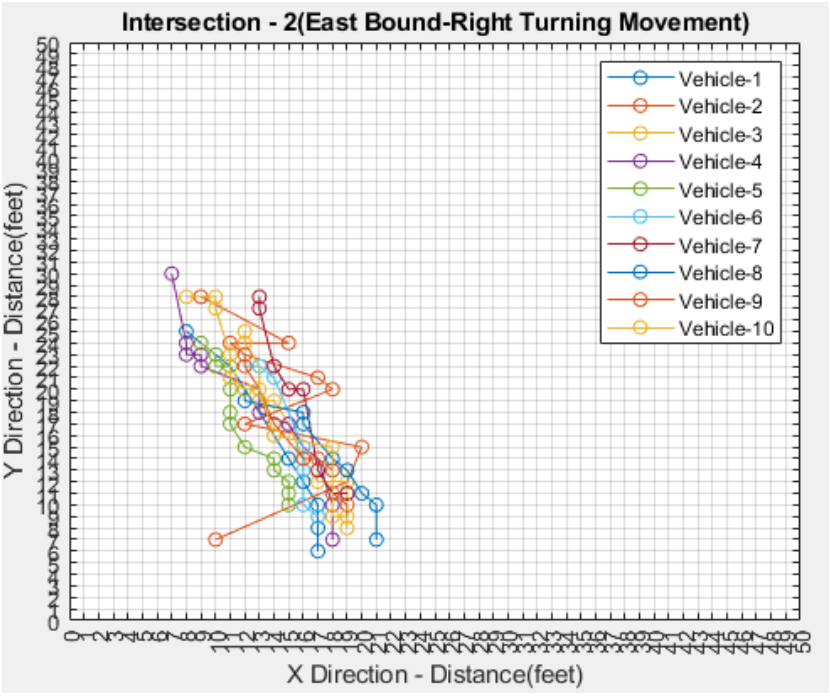


(f) South Bound (SB) Through

Figure E.1 (Continued) Sample Trajectories from Intersection – 2.

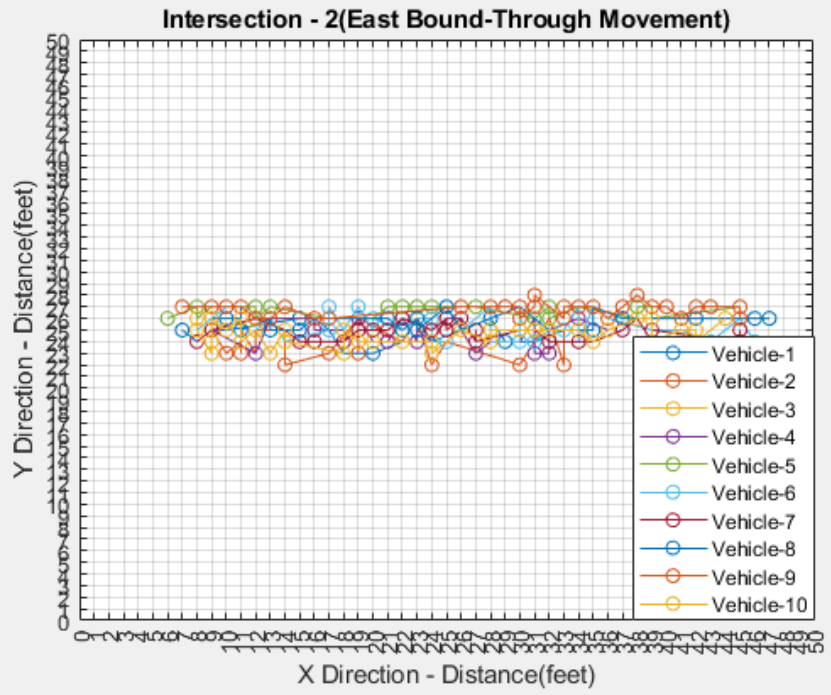


(g) East Bound (EB) Left Turn

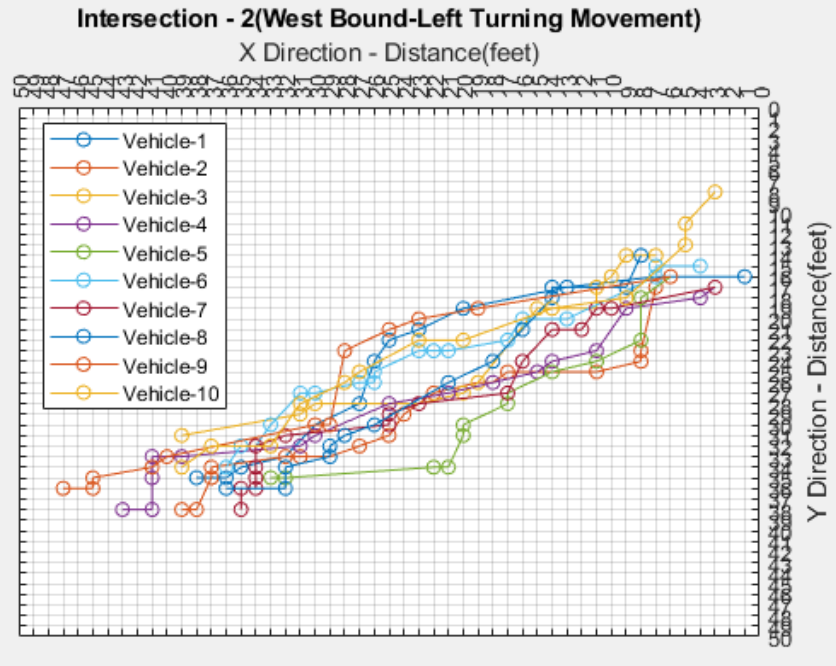


(h) East Bound (EB) Right

Figure E.1 (Continued) Sample Trajectories from Intersection – 2.

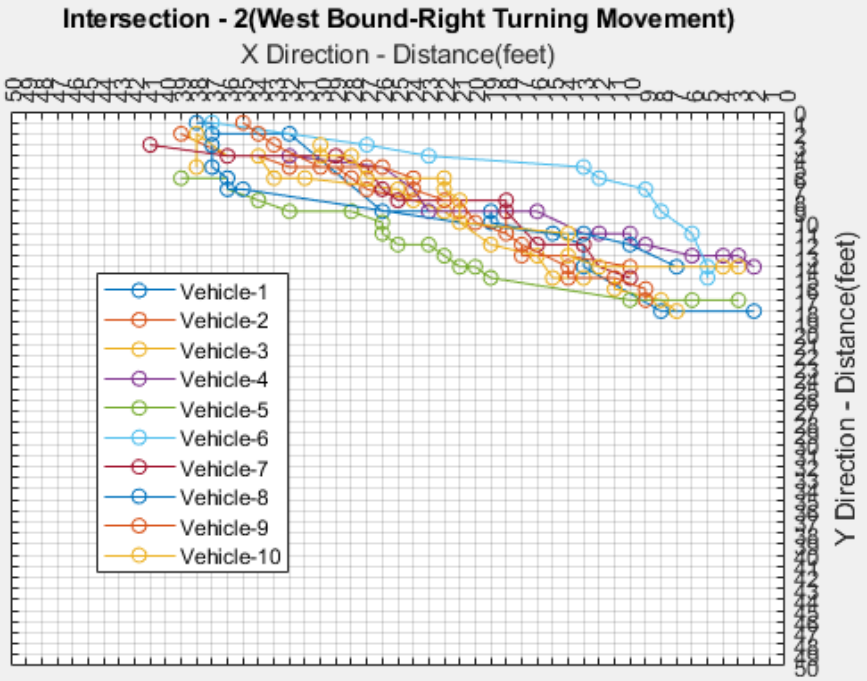


(i) East Bound (EB) Through

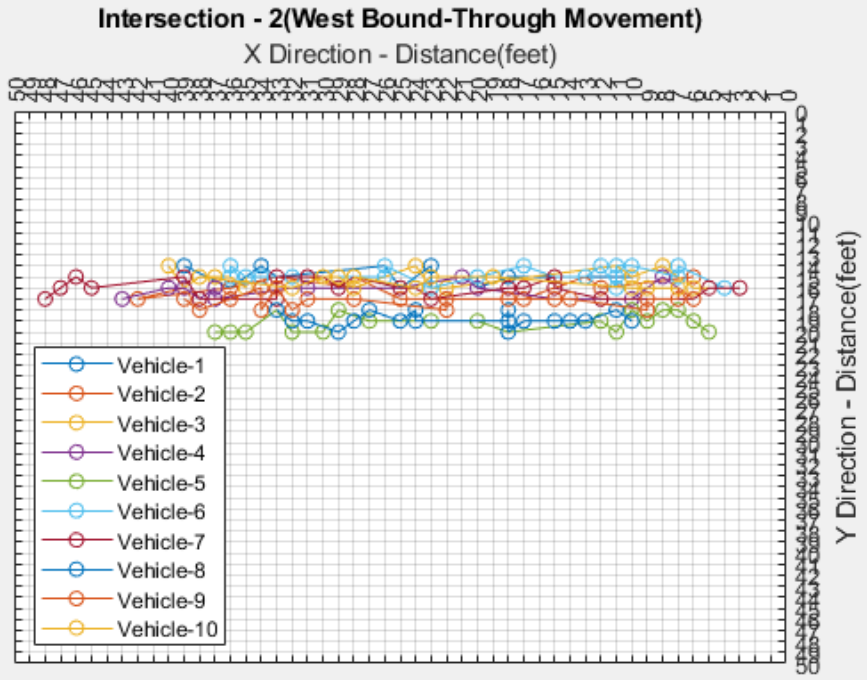


(j) West Bound (WB) Left Turn

Figure E.1 (Continued) Sample Trajectories from Intersection – 2.



(k) West Bound (WB) Right



(i) West Bound (WB) Through

Figure E.1 (Continued) Sample Trajectories from Intersection – 2.

REFERENCES

1. United Nations, Department of economic and social affairs, “World Urbanization Prospects”, ISBN 978-92-1-151517-6, 2014. Available: https://www.un.org/en/development/desa/publications/2014-revision-world-urbanization-prospects.html. [Accessed on: 11/24/2018]

2. United Nations, U.S. Department of Commerce, “Southern Cities Growing Quickly”, 2016. Available: https://www.census.gov/library/visualizations/2017/comm/cb17-81-cities-growing.html. [Accessed on: 11/30/2018]

3. United States Census Bureau, “2010 Census Urban and Rural Classification and Urban Area Criteria”. Available: https://www.census.gov/geo/reference/ua/urban-rural-2010.html. [Accessed Jan. 18, 2018].

4. U.S. Department of Transportation, “Smart City Challenge”. Available: https://www.transportation.gov/smartcity. [Accessed Jan 15, 2018].

5. Smart America, “Smart Cities USA”. Available: http://smartamerica.org/teams/smart-cities-usa/. [Accessed on Jan 15, 2018].

6. Business Insider Intelligence, “There will be 24 Billion IoT Devices Installed on Earth by 2020”. Available: http://www.businessinsider.com/there-will-be-34-billion-iot-devices-installed-on-earth-by-2020-2016-5. [Accessed Jan 18, 2018].

7. J. Neira, J. D. Tardos, J. Horn and G. Schmidt, "Fusing Range and Intensity Images for Mobile Robot Localization", *IEEE Transactions on Robotics and Automation,* vol. 15, no. 1, pp. 76-84, Feb. 1999.

8. M. Sergi, C. Shankwitz and M. Donath, "LIDAR-Based Vehicle Tracking for a Virtual Mirror", *IEEE IV Intelligent Vehicles Symposium*, Cat. No.03TH8683, Columbus, OH, USA, pp. 333-338, 2003.

9. A. Mendes, L. C. Bento and U. Nunes, "Multi-Target Detection and Tracking with a Laser Scanner", *IEEE Intelligent Vehicles Symposium*, Parma, Italy, pp. 796-801, 2004.

10. A. von Reyher, A. Joos and H. Winner, "A Lidar-Based Approach for Near Range Lane Detection", *IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, pp. 147-152, 2005.

11. T. Ogawa and K. Takagi, "Lane Recognition Using On-vehicle LIDAR", *IEEE Intelligent Vehicles Symposium*, Tokyo, pp. 540-545, 2006.

12. S. Kammel and B. Pitzer, "Lidar-Based Lane Marker Detection and Mapping", *IEEE Intelligent Vehicles Symposium*, Eindhoven, pp. 1137-1142, 2008.

13. P. Lindner, E. Richter, G. Wanielik, K. Takagi and A. Isogai, "Multi-Channel Lidar Processing for Lane Detection and Estimation", *12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, MO, pp. 1-6, 2009.

14. S. Shin, I. Shim and I. S. Kweon, "Combinatorial Approach for Lane Detection Using Image and LIDAR Reflectance", *12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Goyang, pp. 485-487, 2015.

15. J. Wu, H. Xu and J. Zheng, "Automatic Background Filtering and Lane Identification with Roadside LiDAR Data", *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, pp. 1-6, 2017.

16. G. Brzezinska, A. Dorota, “Airborne LiDAR: A New Source of Traffic Flow Data”, *Ohio State University, Dept. of Civil & Environmental Engineering & Geodetic Science*, TE716.O3 A37 no. 05-014; HE336.T7 G74 2005x OTP 1.23/7:2005-14, 2005.

17. H. Zhao, X. W. Shao, K. Katabira and R. Shibasaki, "Joint Tracking and Classification of Moving Objects at Intersection Using a Single-Row Laser Range Scanner", *IEEE Intelligent Transportation Systems Conference*, Toronto, Ontario, pp. 287-294, 2006.

18. Yao, Wenqing, “LIDAR-Based Vehicle Tracking for Stopping Distance Measurement at Intersections”, 2012. Available: https://www.google.com/search?client=firefox-b-1-d&q=LIDAR-Based+Vehicle+Tracking+for+Stopping+Distance+Measurement+at+Intersections. [Accessed on: 3/24/2020]

19. A. Fod, A. Howard and M. A. J. Mataric, "A Laser-Based People Tracker", *IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), Washington, DC, USA, pp. 3024-3029 vol.3, 2002.

20. Huijing Zhao and R. Shibasaki, "A Novel System for Tracking Pedestrians Using Multiple Single-Row Laser-Range Scanners", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 2, pp. 283-291, 2005.

21. J. Cui, H. Zha, H. Zhao, R. Shibasaki, "Laser-Based Detection and Tracking of Multiple People in Crowds", *Computer Vision and Image Understanding*, Volume 106, Issues 2–3, pp. 300-312, ISSN 1077-3142, 2007.

22. F. Nashashibi and A. Bargeton, "Laser-Based Vehicles Tracking and Classification Using Occlusion Reasoning and Confidence Estimation", *IEEE Xplore*, 2008.

23. U. Ramer. "An Iterative Procedure for The Polygonal Approximation of Plane Curves", *Computer Graphics and Image Processing*, Volume 1, Issue 3, pp. 244-256, ISSN 0146-664X, 1:224–256, 1972.

24. R. Yang. "Vehicle Detection and Classification from a LIDAR equipped probe vehicle", *Ohio state university*- M.S Thesis, 2009.

25. D. A. Thorton, K. Redmill, B. Coifman, "Automated Parking Surveys from a LIDAR Equipped Vehicle", *Transportation Research Part C*, pp. 23–35, 2014.

26. F. Zhang, D. Clarke, and A. Knoll, "LiDAR Based Vehicle Detection in Urban Environment", *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI),* Beijing pp. 1-5, 2014.

27. I. Yani et al. "Difference of Normals as a Multi-scale Operator in Unorganized Point Clouds", *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission,* pp. 501-508, 2012.

28. M. A. Hearst. "Support Vector Machine", *University of California, Berkeley.* Available: http://alex.smola.org/papers/2002/SchSmo02b.pdf. [Accessed on: 2/7/2018].

29. M. Thuy, F. Puente León, "Non-Linear Multimodal Object Tracking Based on 2D LiDAR Data", *Metrology and measurement systems*, Metrol. Meas. Syst. Vol. XVI, No 3, pp. 359-369, Index 330930, ISSN 0860-8229, 2009.

30. T. Taipalus and J. Ahtiainen, "Human Detection and Tracking with Knee-High Mobile 2D LIDAR," *2011 IEEE International Conference on Robotics and Biomimetics*, Karon Beach, Phuket, pp. 1672-1677, 2011.

31. A. P. Tarko, K. B. Ariyur, M. A. Romero, and V. K. Bandaru, “T-Scan: Stationary LiDAR for Traffic and Safety Applications - Vehicle Detection and Tracking”, *Indiana Department of Transportation and Purdue University,* Technical Report-August 2016 DOI: 10.5703/1288284316347.

32. B. Kluge, C. Kohler and E. Prassler, "Fast and Robust Tracking of Multiple Moving Objects with a Laser Range Finder", *IEEE International Conference on Robotics and Automation* (Cat. No.01CH37164), Seoul, South Korea, pp. 1683-1688 vol.2, 2001.

33. K. Ruohonen, “Graph Theory”. Available: http://math.tut.fi/~ruohonen/GT_English.pdf. [Accessed on: 1/1/2019].

34. J. Salvatore. “Bipartite Graphs and Problem Solving”. University of Chicago. Available: https://math.uchicago.edu/~may/VIGRE/VIGRE2007/REUPapers/INCOMING/Jimmy%20Salvatore%20Graph%20Theory%202.pdf. [Accessed: 12/20/2018].

35. A.V. Ovanesova, L.E. Suárez, “Applications of Wavelet Transforms to Damage Detection in Frame Structures”, *Engineering Structures*, Volume 26, Issue 1, pp. 39-49, ISSN 0141-0296, 2004.

36. M. Rucka, K. Wilde, “Application of Continuous Wavelet Transform in Vibration-Based Damage Detection Method for Beams and Plates, Journal of Sound and Vibration”, Volume 297, Issues 3–5, pp. 536-550, ISSN 0022-460X, 2006.

37. M. Solís, M. Algaba, P. Galvín, “Continuous Wavelet Analysis of Mode Shapes Differences for Damage Detection”, *Mechanical Systems and Signal Processing*, Volume 40, Issue 2, pp. 645-666, ISSN 0888-3270, 2013.

38. W. J. Rakowski, “Wavelet Approach to Damage Detection of Mechanical Systems and Structures”, *Procedia Engineering*, Volume 182, pp. 594-601, ISSN 1877-7058, 2017.

39. X. Jiang, A. H, “Wavelet Packet-Autocorrelation Function Method for Traffic Flow Pattern Analysis”, *Computer-Aided Civil & Infrastructure Engineering,*19(5):324-3372004.

40. D. B. Giralda, F et al., “Wavelet-Based Denoising for Traffic Volume Time Series Forecasting with Self-Organizing Neural Networks”, *Computer-Aided Civil and Infrastructure Engineering*, 25 (7), pp. 530-545, 2010.

41. Z. Zheng, S. Ahn, D. Chen, J. Laval, “Applications of Wavelet Transform for Analysis of Freeway Traffic: Bottlenecks, Transient Traffic, and Traffic Oscillations”, *Transportation Research Part B: Methodological*. Volume 45, Issue 2, pp. 372-384, ISSN 0191-2615, 2011.

42. D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels and P. Jaillet, "Wavelets on Graphs with Application to Transportation Networks," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, pp. 1707-1712, 2014.

43. T. Lu. “Lecture 12: Discrete Laplacian - Discrete Laplacian”, *Stanford University*. Available: https://graphics.stanford.edu/courses/cs468-13spring/assets/lecture12-lu.pdf. [Accessed on: 1/25/2019].

44. L. Zhao, J. Lee, S. Chien, and C. Oh, “Shockwave-Based Automated Vehicle Longitudinal Control Algorithm for Nonrecurring Congestion Mitigation”. *Hindawi, Journal of Advanced Transportation*, Article ID 6568135, 2017.

45. L. Matthies and A. Elfes, "Integration of Sonar and Stereo Range Data Using a Grid-Based Representation", *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, USA, pp. 727-733 vol.2, 1988.

46. T. Vu, O. Aycard and N. Appenrodt, "Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments", *IEEE Intelligent Vehicles Symposium*, Istanbul, pp. 190-195, 2007.

47. M. E. Bouzouraa and U. Hofmann, "Fusion of Occupancy Grid Mapping and Model-Based Object Tracking for Driver Assistance Systems Using Laser and Radar Sensors", *IEEE Intelligent Vehicles Symposium*, San Diego, CA, pp. 294-300, 2010.

48. R. H. Wong, J. Xiao and S. L. Joseph, "A Robust Data Association for Simultaneous Localization and Mapping in Dynamic Environments", *The IEEE International Conference on Information and Automation*, Harbin, pp. 470-475, 2010.

49. J. Moras, V. Cherfaoui and P. Bonnifait, "Credibilist Occupancy Grids for Vehicle Perception in Dynamic Environments", *IEEE International Conference on Robotics and Automation, Shanghai*, pp. 84-89, 2011.

50. M. Schütz, N. Appenrodt, J. Dickmann and K. Dietmayer, "Occupancy Grid Map-Based Extended Object Tracking", *IEEE Intelligent Vehicles Symposium*, Dearborn, MI, pp. 1205-1210, 2014.

51. S. E. Hadji, T. H. Hing, M. S. M. Ali, M. A. Khattak and S. Kazi, "2D Occupancy Grid Mapping with Inverse Range Sensor Model", *10th Asian Control Conference (ASCC)*, Kota Kinabalu, pp. 1-6, 2015.

52. R. Dia, J. Mottin, T. Rakotovao, D. Puschini, S. Lesecq, “Evaluation of Occupancy Grid Resolution through a Novel Approach for Inverse Sensor Modeling”, *IFAC-Papers Online*, Volume 50, Issue 1, pp. 13841-13847, ISSN 2405-8963, 2017.

53. B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung and J. W. Choi, "Probabilistic Vehicle Trajectory Prediction Over Occupancy Grid Map via Recurrent Neural Network", *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, pp. 399-404, 2017.

54. R. Weston, S. Cen, P. Newman and I. Posner, "Probably Unknown: Deep Inverse Sensor Modelling Radar", *International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, pp. 5446-5452, 2019.

55. S. Steyer, G. Tanzmeister and D. Wollherr, "Grid-Based Environment Estimation Using Evidential Mapping and Particle Tracking", *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 384-396, 2018.

56. C. G. Prevost, A. Desbiens and E. Gagnon, "Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle", *American Control Conference*, New York, NY, pp. 1805-1810, 2007.

57. V. Subramanian, T. F. Burks, W. E. Dixon, “Sensor Fusion Using Fuzzy Logic Enhanced Kalman Filter for Autonomous Vehicle Guidance in Citrus Groves”, *American Society of Agricultural and Biological Engineers*, ISSN 0001-2351, 2009.

58. X. Li, K. Wang, W. Wang and Y. Li, "A Multiple Object Tracking Method Using Kalman Filter”, *IEEE International Conference on Information and Automation*, Harbin, pp. 1862-1866, 2010.

59. A. Salarpour, A. Salarpour, M. Fathi, & M. H. Dezfoulian, “Vehicle Tracking Using Kalman filter and Features”, *An International Journal (SIPIJ),* Volume 2. Available: https://www.researchgate.net/publication/270018823_Vehicle_Tracking_Using_Kalman_Filter_and_Features. [Accessed on: 11/25/2019]

60. C. Barrios and Y. Motai, "Improving Estimation of Vehicle's Trajectory Using the Latest Global Positioning System with Kalman Filtering", *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 12, pp. 3747-3755, 2011.

61. S. E. Li, G. Li, J. Yu, C. Liu, B. Cheng, J. Wang, K. Li, “Kalman Filter-Based Tracking of Moving Objects Using Linear Ultrasonic Sensor Array for Road Vehicles*”, Mechanical Systems and Signal Processing*, Volume 98, pp. 173-189, ISSN 0888-3270, 2018.

62. Y. Maalej, S. Sorour, A. Abdel-Rahim and M. Guizani, "Tracking 3D LIDAR Point Clouds Using Extended Kalman Filters in KITTI Driving Sequences", *IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, pp. 1-6, 2018.

63. H. Veeraraghavanm and N. Papanikolopoulos, "Combining Multiple Tracking Modalities for Vehicle Tracking at Traffic Intersections", *IEEE International Conference on Robotics and Automation* (ICRA), New Orleans, LA, USA, pp. 2303-2308 Vol.3, 2004.

64. E. Bas, A. M. Tekalp and F. S. Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis", *IEEE Intelligent Vehicles Symposium*, Istanbul, pp. 392-397, 2007.

65. A. Asvadi, P. Girão, P. Peixoto and U. Nunes, "3D Object Tracking Using RGB and LIDAR Data", *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, pp. 1255-1260, 2016.

66. X. Mi, H. Ren, Z. Ouyang, W. Wei, K Ma*, “*The Use of The Mexican Hat and The Morlet Wavelets for Detection of Ecological Patterns”, *Plant Ecology 179*, pp. 1–19, 2005.

67. J. Wang, K. Dixon, H. K. Li, & J. Ogle, “Normal Acceleration Behavior of Passenger Vehicles Starting from Rest at All-Way Stop-Controlled Intersections”, *Transportation Research Record,* 1883(1), pp. 158–166.

68. Scanse User’s Manual and Technical Specifications-sweep v1.0. Available: https://s3.amazonaws.com/scanse/Sweep_user_manual.pdf. [Accessed on: 8/10/2018].

69. K. C. David, “Evaluation of the Accuracy of Traffic Volume Counts Collected by Microwave Sensors”, *Brigham Young University,* BYU Scholars Archive, 2015.

70. D. Middleton, and R. Parker, “Vehicle Detector Evaluation”, *Texas Department of Transportation, The Texas A&M University System, U.S. Department of Transportation, Washington*, *D.C.*, 2002. Available: https://pdfs.semanticscholar.org/89f2/ee6cb31106a543d91335ae4a6eeb59b870bf.pdf. [Accessed on: 9/11/2019]

71. P. Bellucci, and C. Ernesto, “Data Accuracy on Automatic Traffic Counting: The SMART Project Results”. *European Transport Research Review*, Vol. 2, pp. 175–187, 2010.

72. T. M. Peter, G. Dharmavaram, and A. Stevanovic. “Evaluation of UDOT’S Video Detection Systems”, System’s Performance in Various Test Conditions, Mountain-Plains Consortium, Salt Lake City, Utah, 2004.

73. IEEE spectrum, “Sweep Is a $250 LIDAR with Range of 40 Meters That Works Outdoors”. Available: https://spectrum.ieee.org/automaton/robotics/robotics-hardware/sweep-lidar-for-robots-and-drones. [Accessed on: 3/24/2020]

74. Raspberry pi forum. Available: https://www.raspberrypi.org/. [Accessed on: 4/25/2020]

75. Hyper Juice product. Available: https://www.hypershop.com/collections/vendors?q=hyperjuice. [Accessed on: 4/26/2020]

76. M. Marti, R. Kuehl, S. Petersen, “Traffic Data Collection Improvements”, *Transportation Research Services Section*, SRF Consulting Group, Inc. and Minnesota Department of, Report No: MN/RC 2014RIC51B, 2014.

77. S. Swann, “Albury City Comparison of Traffic Data Collection Methods”, Albury City, New South Wales, Australia, 2010. Available: http://miovision.com/wp-content/uploads/AlburyCity_Comparison_of_Traffic_Data_Collection_Methods_October2010.pdf. [Accessed on 5/16/2020].

78. Traffic Data Accuracy Definition, miovision. Available: https://help.miovision.com/s/article/Traffic-Data-Accuracy-Definition?language=en_US. [Accessed on 6/8/2020]

79. H. Xu, Z. Tian, J. Wu, H. Liu, and J. Zhao, “High-Resolution Micro Traffic Data from Roadside LiDAR Sensors for Connected-Vehicles and New Traffic Applications”, *Nevada Department of Transportation* Research *Report,* No. 224-14-803 TO 15, 2018.

80. R. Yang, “Vehicle Detection and Classification from a LIDAR equipped probe vehicle”, *Electrical and Computer Engineering - Ohio State University,* 2009.

81. V. K. Bandaru, “Algorithms for LiDAR Based Traffic Tracking: Development and Demonstration”, *Purdue University*, 2016. Available: https://docs.lib.purdue.edu/open_access_theses/922. [Accessed on: 6/8/2020]

82. M. Sualeh, G.-W. Kim, “Dynamic Multi-LiDAR Based Multiple Object Detection and Tracking”, *Sensors*, *19*, 1474, 2019.

83. E and M Robotics, “Multi-Layer LiDAR Solutions That Fit Your Application”. Available: https://www.eandm.com/Robotics/Robotics.aspx#3dLiDar. [Accessed: 6/10/2020].

84. K. Korosec, “Waymo to start selling standalone LiDAR sensors”, *Tech Crunch,* 2019. Available: https://techcrunch.com/2019/03/06/waymo-to-start-selling-standalone-lidar-sensors/. [Accessed: 6/11/2020]