Safety and Operations of Large-Area Rural/Urban Intermodal Systems

SOLARIS

A Tier 1 university transportation center led by the University of Nevada, Reno.

Final Report

October 2018

HIGH-RESOLUTION MICRO TRAFFIC DATA FROM ROADSIDE LIDAR SENSORS FOR CONNECTED-VEHICLES AND NEW TRAFFIC APPLICATIONS

SOLARIS Consortium, Tier 1 University Transportation Center

Center for Advanced Transportation Education and Research
Department of Civil and Environmental Engineering
University of Nevada, Reno
Reno, NV 89557

Hao Xu, P.E., Ph.D.
Zong Tian, P.E., Ph.D.
Jianqing Wu
Center for Advanced Transportation Education and Research
University of Nevada, Reno

Hongchao Liu, P.E., Ph.D.
Junxuan Zhao
Texas Tech University

***DISCLAIMER*:**

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

ACKNOWLEDGMENT OF SPONSORSHIP

This work was sponsored by the SOLARIS Institute, a Tier 1 University Transportation Center (UTC) under Grant No. DTRT13-G-UTC55 and the Nevada Department of Transportation (NDOT) under Grant No. P224-14-803/TO #15.
This research was also supported by engineers at NDOT, the Regional Transportation Commission of Washoe County, Nevada, and the City of Reno.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

AUTHOR ACKNOWLEDGMENTS

The research reported herein was performed under SOLARIS PROJECT DTRT13-G-UTC55 and NDOT PROJECT P224-14-803/TO #15 by the Department of Civil and Environmental Engineering at the University of Nevada, Reno (UNR) and the Department of Civil, Environmental, and Construction Engineering at the Texas Tech University (TTU). UNR was the contractor for this study, with the collaboration with TTU.

Dr. Hao Xu, P.E., Assistant Professor of Civil Engineering at UNR, was the Project Principal Investigator. The other authors of this report are Dr. Zong Tian, Professor of Civil Engineering at UNR and co-Principal Investigator; Jianqing Wu, Research Assistant and Ph.D. Candidate at UNR; Dr. Hongchao Liu, Professor of Civil Engineering at TTU; Junxuan Zhao, Research Assistant and Ph.D. Candidate at TTU.

ABSTRACT

This report documents and presents algorithms and procedure developed for extracting high-accuracy high-resolution trajectory data from roadside LiDAR sensors. The developed methods were evaluated with data from various traffic scenarios. Pilot applications of roadside LiDAR trajectory data for pedestrian-crossing-road prediction, animal-crossing-road detection, and near-crash events identification were also included in this report. The roadside-LiDAR data-processing procedure includes new algorithms of LiDAR-data background filtering, LiDAR-point clustering, cluster classification (vehicles and pedestrians), object tracking and trajectory calculation. The methods for processing roadside LiDAR and pilot applications of using LiDAR trajectory data will serve as a foundation for new connected/autonomous traffic infrastructure advanced by 360-degree edge LiDAR sensors. Road-side LiDAR is new technology to fill the data gap of unconnected multimodal traffic in connected and autonomous traffic systems and will innovate traffic engineering/research areas with all-traffic trajectory data that was not available in traditional traffic sensing systems.

EXECUTIVE SUMMARY

Existing traffic data includes flow rates, occupancy, average speed, and spot speed. Yet new traffic systems and applications require traffic flow information with more detail and higher accuracy—specifically, *multimodal, all-traffic trajectories*. All-traffic trajectory data is critical to various traffic research/engineering areas:

- **Connected and autonomous vehicles (CAVs):** At present, independent onboard sensing systems do not provide enough information for safe operation in multimodal traffic. An autonomous vehicle with advanced sensors could still be hit by another car on a cross street that fails to stop. For advancing traffic safety, vehicles need to obtain trajectories of all traffic in extended distances so they can "detect" traffic changes and risks around corners.
- **Near-crash analysis:** Near-crash events provide essential data for proactive safety analysis and countermeasure recommendation, but this data is difficult to obtain. If all-traffic trajectory data could be collected, we could study vehicle interactions at multiple scales, and define and extract near-crash events to identify traffic safety issues and recommend countermeasures.
- **Traffic performance evaluation/adaptive traffic signal control:** All-traffic trajectories provide comprehensive information to evaluate traffic performance. Trajectory data reports *each road user's* stop location, stop time, speed change, and interaction with other road users in addition to conventional vehicle-traffic performance indices such as the number of stops, delay, travel time, and queue length. Optimizing signals along a road is challenging using conventional traffic sensors because system details cannot be accurately observed. Real-time, all-traffic trajectory data can make the traffic system completely observable, thus revolutionizing adaptive traffic control and outperforming conventional systems.
- **Automatic pedestrian/wildlife-crossing warning signals:** An important application of real-time, all-traffic trajectories is monitoring and predicting vehicle-pedestrian conflicts on urban roads or vehicle-wildlife collision risks on rural highways. Most conventional automatic pedestrian/wildlife warning systems rely on predefined detection areas. These systems trigger warning signals whenever an object is detected in the sensing area, but this has both caused false alarms and failed to identify risks outside the defined areas. Trajectory data tracks the continuous movement of each road user, so crossing detection and prediction can be based on historical trajectory and real-time direction/speed/location for superior accuracy and reliability.

Existing Intelligent Transportation System (ITS) sensors such as loop detectors, video detectors, and Bluetooth sensors provide macro traffic data such as traffic flow rates, average speeds, and occupancy. Given that existing traffic sensors do not provide trajectory data, 360-degree light detection and ranging (LiDAR) sensors are a new option because they detect surrounding objects with high accuracy and frequency and are not influenced by light conditions. The project team developed algorithms specifically for roadside LiDAR sensing systems. LiDAR data processing and high-resolution trajectory extraction are the base function of LiDAR-enhanced traffic infrastructure. Due to sensor installation and data characteristics, methodologies for roadside LiDAR data processing are different from the methods for

autonomous vehicles (Figure 1).

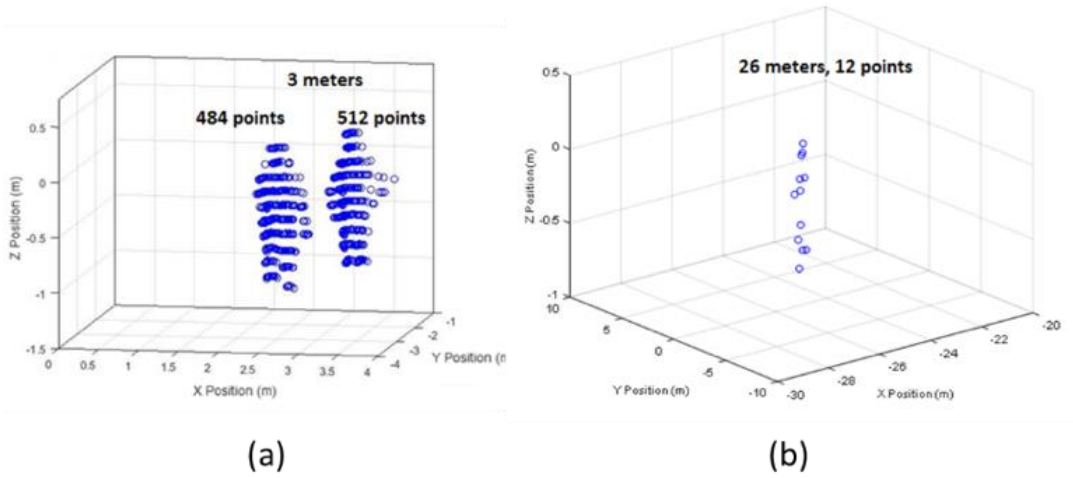


Figure 0-1 Different densities of pedestrian LiDAR points for onboard CAV sensing (a) and roadside surveillance (b) based on detection range requirements. Roadside data processing is much more challenging in this regard.

The project team has developed a procedure for roadside LiDAR data processing, including major steps of background, object clustering, identification of road user types, tracking road users in different data frames, and output of traffic trajectory data (Figure 2). Artificial neural network and support vector machine algorithms have also been applied by the research team to identify and track objects. The characteristics of different road users' LiDAR points are used to determine the types; the features for distinguishing different objects include distance to the sensor, the number of LiDAR points, and the spatial distribution of the cloud points.

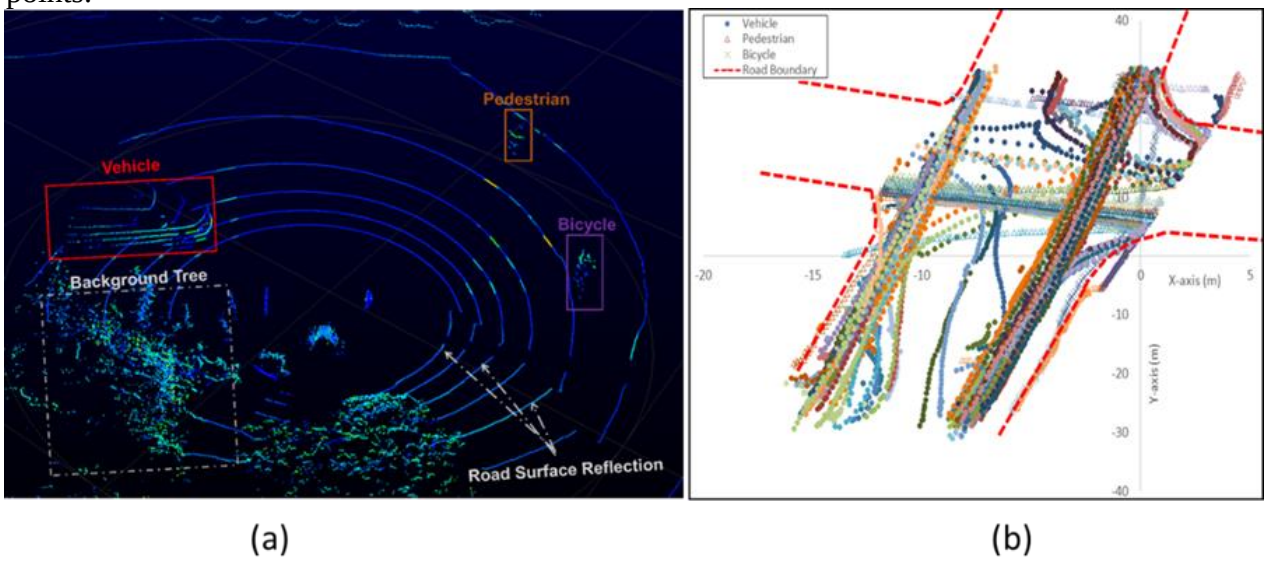


Figure 0-2 (a) Unprocessed LiDAR frame from a roadside Velodyne VLP-16 sensor, showing how resulting point-cloud data fix and can track objects precisely in three dimensions; (b) high-accuracy, multimodal traffic trajectories obtained using sequences of point cloud data from a roadside LiDAR sensor by the project team

The developed background-filtering algorithm filtered more than 99% of background points when there were no pedestrians or vehicles (case study at N Virginia St and 15th St). Even at the site with a high pedestrian and vehicle volume (case study at N Virginia St and 10th St), the method still removed more than 97% of background points. The detection accuracy of object clustering was about 96%. The main reasons causing clustering failure included sparse points from objects at a far distance and occlusion. An approximate 96% classification accuracy can be guaranteed within the 30-meter-radius sensing range. The classification failure was found to be caused by vehicle occlusion. Tracking accuracy of approximately 95% can be achieved within about 30 meters detection range from the LiDAR sensor. Similar reasons have been found for tracking failure as those addressed in the clustering section. The effective distance was for VLP-16 LiDAR sensors; the project team achieved doubled effective-distance with VLP-32 LiDAR sensors. The statistic results showed that 90% of the speed values had error lower than 2.5 mph.

For pilot applications, the average accuracy of pedestrian-crossing prediction was 97% and non-crossing prediction accuracy was 84%. The average detection range was 30.72 m from the VLP16 LiDAR sensor. The max effective detection range of the data processing procedure was 37.74 m. Though deer could be seen in the LiDAR data visualization at 37.74 m,, the developed algorithms could not successfully identify these points as deer. The near-crash identification method was successfully applied for extraction of near-crash events at two case-study sites.

INTRODUCTION

New-generation transportation systems employ advanced sensing and communication technologies to sense the surrounding traffic environment and exchange real-time information between road users and traffic infrastructures, such as autonomous vehicles (AVs) and connected vehicles (CVs). Various sensors of an AV work together to sense surrounding objects including roads, other road users, and obstacles. The sensors' detection distances and occlusion scenarios limit an AV's sensing range, so CV technologies are needed to enhance an individual vehicle's sensing capability. CV systems communicate real-time movement status between road users and broadcast real-time traffic information from traffic infrastructure. A CV can know location, speed, and direction of another CV if they communicate with each other. CV technologies allow extended distance for connected drivers or AVs to "see" around corners or through obstacles by shared information. Safety threats and traffic changes can be perceived in an extended range so that we can avoid collisions and reduce travel time. However, the full benefits of connected-vehicle applications need all road users to be equipped with communication devices and exchange information with others. The mixed traffic including both connected-vehicles and unconnected-vehicles will exist in the next decades or even longer, but existing CV systems cannot perceive unconnected road users that do not communicate. The data gap of unconnected road users will limit the actual benefits of CAVs, create public mistrust in connected-traffic systems, and hinder deployment of CAVs. Accuracy and reliability of exchanged information can also be influenced by communication reliability and security, so the data issue will persist even when all traffic components are connected. Sensing all-traffic trajectories, especially unconnected road users, is essential for connected-traffic applications that can also advance traffic safety and mobility. For example, trajectories of vehicles and pedestrians can be used to analyze near-crash events for safety improvement and to evaluate signal performance with much more traffic details than traditional data.

High-resolution micro traffic data (trajectory data) can be obtained by conventional probe vehicles with Global Positioning System (GPS) devices. However, probe vehicles provide only sample trajectories, while CV applications require data of all road users. Existing infrastructure-mounted sensors such as inductive loop, cameras, Bluetooth sensors, and radar sensors provide macro traffic data of traffic flow rates, average speeds, and occupancy, which still do not meet the requirement of CV applications. Besides, the illumination condition has a significant effect on the video quality and performance of camera sensors (Mukhtar et al., 2015) and existing radar sensors mostly give the spot speed and traffic volume only. In this regard, application of road-edge light-detection-and-ranging (LiDAR) sensors can fill the data gap of multimodal trajectory data by shooting pulsed-laser beams to measure object positions accurately (Csanyi and Toth, 2007). 360-degree LiDAR sensors detect the surrounding environment at high accuracy and high resolution without an influence of light condition. During each 360-degree rotation, the sensor collects a cloud of surface points of surrounding objects at a centimeter-level accuracy and generates a LiDAR data frame as demonstrated in Figure 1-1. The cloud points from roadside LiDAR can be used to identify classifications, and locations of objects, so continuous LiDAR data frames offer an opportunity to monitor trajectories of all road users.

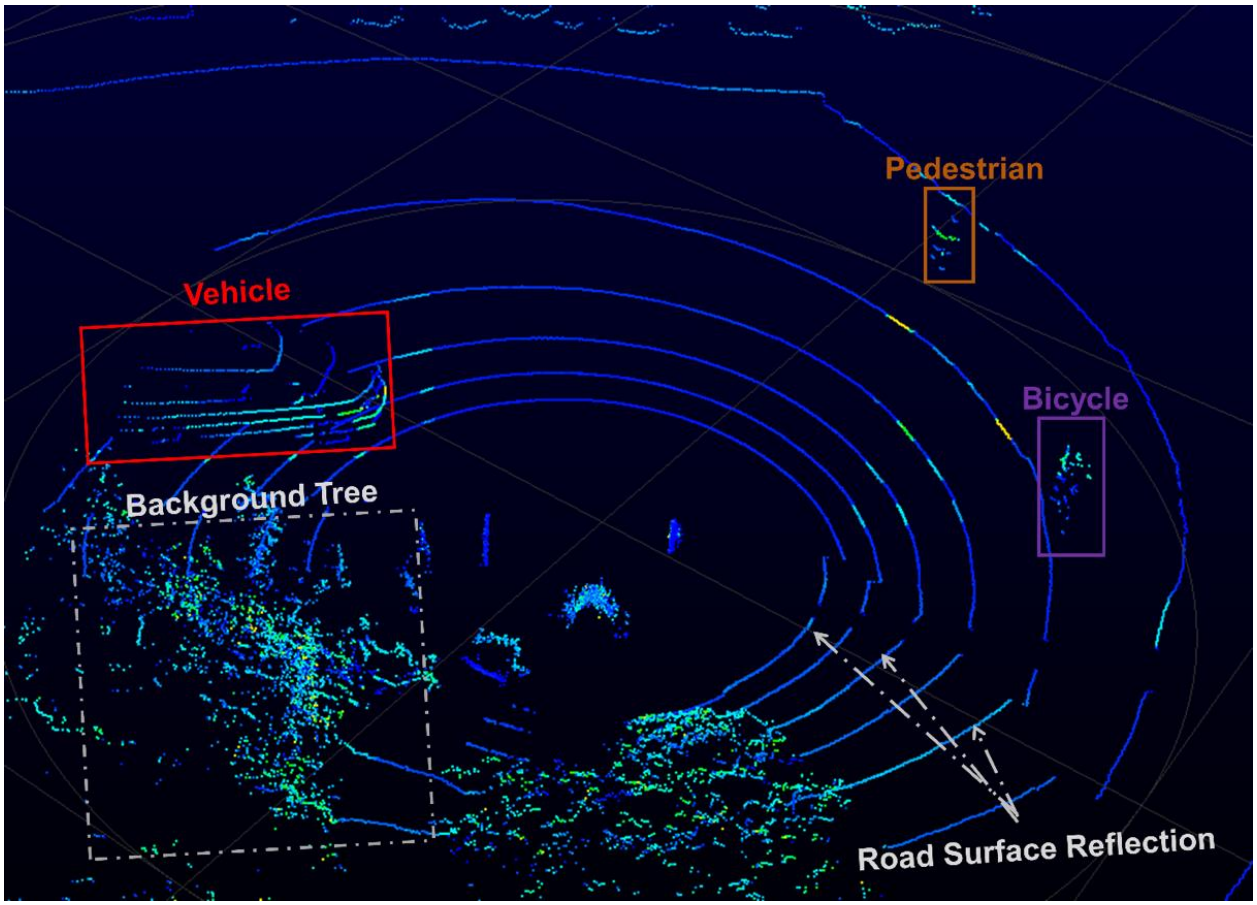


Figure 0-4 Demonstration of a 360-degree LiDAR data frame

Although road-edge LiDAR sensors could offer trajectory data, roadside deployment of LiDAR had been economically infeasible until recently when LiDAR sensor prices dramatically dropped. LiDAR sensors have been widely used for surveying and mapping; recent applications are for robots and autonomous vehicles to sense the surrounding environment and objects (Reina et al., 2011). Existing data-processing algorithms for those applications could not directly be used to process roadside data because of different data characteristics, environment, and deployment requirements, so roadside LiDAR data processing methodologies and roadside-LiDAR applications need to be researched for different traffic scenarios. Roadside LiDAR can serve CAV systems by generating all-traffic trajectories and broadcasting them through dedicated-short-range-communication (DSRC) (Zeng et al., 2009). Trajectory data from roadside LiDAR can also be used to revolutionize conventional traffic engineering areas of safety, mobility, and modeling. Both connected and unconnected road users will benefit from roadside LiDAR technologies.

This project developed a procedure, including multiple algorithms, for generating high-accuracy multimodal traffic trajectories with roadside 360-degree LiDAR sensors. The developed data-processing procedure first excludes points of background objects such as road surface, trees, and poles; then, it clusters left points into objects that are multimodal travelers; it further classifies those objects into pedestrians, vehicles and other road user types. The procedure calculates each road user's location with x-y-z coordinates of clustered points and

estimates their speeds based on time difference and location change in continuous data frames. Finally, trajectories including road user type, location, speed and direction information are obtained. To demonstrate applications of roadside LiDAR, this project applied trajectories from roadside LiDAR for prediction of pedestrians crossing roads, pedestrian-vehicle near-crash analysis, and detection of wildlife animals crossing highways. The major achievements of this project are summarized in the following:

1) Literature review on existing methodologies of detecting and tracking objects with LIDAR
2) Development of an automatic LiDAR background filtering algorithm
3) Development of an algorithm to extract trajectories of road users from roadside LiDAR data
4) Development of an algorithm to identify different road users with roadside LiDAR data
5) Development of an integrated procedure for processing high-resolution cloud points from roadside LiDAR and extracting multimodal traffic trajectories
6) A pilot application of roadside LiDAR to detect and predict pedestrians crossing roads based on LiDAR trajectory data
7) A pilot application of roadside LiDAR to define and extract near-crash events
8) A pilot application of roadside LiDAR to detect wildlife animals crossing a highway

This project report is structured as follows: Chapter 2 presents the literature review results on existing algorithms for processing LiDAR data and other sensor data. Chapter 3 documents the sensor used in this project, the developed data processing procedure, algorithms for all data processing steps, and performance evaluation of the developed methods. Chapter 4 introduces the pilot application of traffic trajectories from roadside LiDAR in the area of pedestrian-crossing prediction, wildlife-crossing detection, and near-crash analysis. Chapter 5 summarized conclusions and recommendations based on findings of this project.

LITERATURE REVIEW

LiDAR has been widely used for remote sensing (Yang et al., 2013). For example, topographic LiDAR uses a near-infrared laser to map the land, and bathymetric LiDAR uses water-penetrating green light to measure seafloor and riverbed elevations. LiDAR technology is also used in sensing systems of AVs to detect road boundaries, pedestrians, vehicles, and obstacles, which is called on-board LiDAR(Li et al., 2004). LiDAR sensors can be classified into two major types: flash LiDAR and rotating LiDAR. Flash LiDAR sensors are also referred to as Time-of-Flight (ToF) camera sensors or ToF LiDAR that scan the scene by a single light pulse to provide high-density laser points in a narrow area. Rotating LiDAR sensors use rotating assemblies or rotating mirrors for laser beams to sweep around and scan surrounding objects in a 360-degree field. To date, there are many LiDAR products from various manufacturers on the market. Table 2-1 lists examples of LiDAR sensors and their manufacturers that were found in the sensor search of this project, while many other LiDAR manufactures and products are not included. The unit prices were received/found during this project, so readers are suggested to contact manufactures for the latest sensors and updated prices with knowing LiDAR prices drop quickly and features are advanced continuously. This project employed Velodyne VLP-16 LiDAR (rotating LiDAR with 16 laser beams) sensors for roadside LiDAR data collection for its relatively low cost and 360-degree lateral field of view.

Although there are limited studies on roadside LiDAR data processing, many algorithms have been developed for airborne and vehicle/robot on-board LiDAR. This chapter reviews existing methodologies in three focused areas: background filtering, object detection (clustering and classification), and object tracking. These methodologies and findings from previous research helped the development of new methods for processing roadside LiDAR data in this project. The feasibility of using existing methods for roadside LiDAR processing was also discussed.

2.1 Background Filtering

In roadside LiDAR data, background objects can be ground surface, buildings, trees and infrastructure that are not road users and interfere object detection/classification. Background filtering removes LiDAR points of those background objects and keeps points of road users including stopped objects. Background filtering serves as the initial step of roadside LiDAR processing and can significantly enhance the accuracy and efficiency of the following steps – object clustering, identification of types and object tracking. Definition of background is different in various LiDAR applications, which is determined by the unique information of each specific application.

Table 0-1 Example LiDAR Sensors from Various Manufacturers

Name	Manufacturer	Detection range	No. of Beams	Frequency	Field of view	Type	Unit price
Vu8	Leddartech	700ft (215m)	8	Up to 100 Hz	20°, 48° and 100°	Flash LiDAR	$650
LeddarOne	Leddartech	130ft (40m)	1	Up to 140 Hz	3°	Flash LiDAR	$115
Leddar M16	Leddartech	325ft(100m)	16	Up to 50 Hz	9º to 95º beam options	Flash LiDAR	$740
Leddar IS16	Leddartech	165ft (50m)	16	Up to 50 Hz	45º	Flash LiDAR	$940
HDL-32E	Velodyne	263-325ft (80-100m)	32	10 Hz	360º	Rotating LiDAR	
HDL-64E	Velodyne	394ft (120m)	64	5-20 Hz	360º	Rotating LiDAR	
VLP-16	Velodyne	325ft(100m)	16	5-20 Hz	360º	Rotating LiDAR	
Puck Lite	Velodyne	325ft(100m)	16	5-20 Hz	360º	Rotating LiDAR	
Puck Hi-Res	Velodyne	325ft(100m)	16	5-20 Hz	360º	Rotating LiDAR	
VUX-1UAV	Riegl	1,150 ft (350m)		200 Hz	330°	Flash LiDAR	$55,000
MiniVUX-1UAV	Riegl	920ft (250m)		100 Hz	360º	Rotating LiDAR	
UAV LidarPod	Routescene	325ft(100m)	32	10 Hz	360º	Rotating LiDAR	
Vx Long Range Lidar Sensor	YellowScan			100 kHz		Flash LiDAR	

Background Filtering for Airborne LiDAR

The background in airborne LiDAR data is normally made up of points of ground surface (Yang et al., 2015). Methods of detecting and filtering background points process evenly-distributed high-density points from airborne LiDAR. Those methods can be grouped into four types: slope-based methods, surface-based methods, segment-based methods, and others. Slope-based methods (Vosselman, 2000) use the erosion-operator formulation of mathematical morphology to define a ground surface with consideration of slope constraints along scan lines. Surface-based ground filtering methods employ an active shape model to identify background surface by minimizing the internal energy (Elmqvist, 2001). Segment-based methods (Yang et al., 2016) first classify laser points into segments (clustered points) and isolated points, then remove all isolated points, and identify the segments of ground surface points with morphological methods. Although methods for filtering ground points of airborne LiDAR data are pretty mature, those algorithms are for evenly-distributed high-density points (output of flash LiDAR rather than rotating LiDAR) and do not consider buildings and vegetation as background. Thus, existing background filtering algorithms for airborne LiDAR cannot be used for roadside LiDAR background filtering.

Background Filtering for On-Board LiDAR

In AV onboard sensing systems, background objects are similar to defined background in roadside sensing systems. However, background points are not directly identified in onboard systems because of background changes along with the movement of vehicles (Brackstone et al., 1999). Onboard LiDAR sensing systems typically use patch segmentation and classification to identify interest objects from the raw LiDAR data (Wang et al., 2012); then, they consider the remaining points as background/noise and exclude them from the following data processing. The segmentation and classification methods normally require more LiDAR points of an object than object points provided by roadside LiDAR sensors that are often cheaper, have fewer laser beams and need to work in an extensive range. Therefore, the existing background filtering methods for on-board sensing systems are not appropriate for roadside LiDAR data processing.

Background Filtering for Roadside LiDAR

Lee and Coifman (2012) developed a method for roadside LiDAR background filtering. It was the only roadside LiDAR background-filtering algorithm found by this literature review. By recognizing that roadside LiDAR sensors return nearly identical scans of the background when there is no vehicle, the authors aggregated multiple no-traffic data frames to identify background objects and removed them from LiDAR data frames. In the field test of this project, it was difficult to obtain enough frames without vehicles at busy traffic sites during the data collection period. Furthermore, the background can be moving objects such as waving trees and bushes. Indicating the location of dynamic background with Lee and Coifman's method was not reliable in some tested scenarios, especially in windy weather.

Background Filtering of Non-LiDAR Data

Many methods were developed for background filtering of non-LiDAR data, especially video data. Typical methods analyze color values of pixels in continuous video/image frames to identify background objects. Background and non-background objects can be separated by the pre-identified pixel color values (mode or mean) (Zheng et al., 2006). These algorithms developed for vision-based data processing could not be used for LiDAR data directly because the roadside LiDAR data are discrete points rather than continuous raster pixels of information.

However, the ideas benefited background filtering of roadside LiDAR data.

The literature review studied common background filtering methods for LiDAR data and non-LiDAR data, but only Lee and Coifman's (2012) method was developed for roadside LiDAR with limitations found in some scenarios. Therefore, a more effective background filtering method is needed to serve roadside-LiDAR data processing in different road environments.

2.2 LiDAR Point Clustering

Various clustering methods were developed for object identification for onboard LiDAR sensing systems. These methods can be roughly divided into four types: centroid-based clustering, hierarchical clustering, distribution-based clustering, and density-based clustering. The representing method of centroid-based clustering is k-means clustering (Morsdorf et al., 2004; Chehata et al., 2008). Hierarchical clustering is also known as connectivity-based clustering. It creates a new cluster with a point that is furthest from the current cluster centroid iteratively; it then merges clusters that are most similar according to chosen similarity or distance measures. Gupta et al. (Gupta et al. 2010) compared the performance of hierarchical tree and k-means using airborne LiDAR data. The results showed that when there were only two groups, the hierarchical method was better than k-means. However, the hierarchical tree failed where more clusters were formed. It needs to be noted that the k-means method requires a predefined number of objects as input. Distribution-based clustering produces complex models that can capture correlation and dependence between attributes. The Gaussian mixture model (GMM) is a typical distribution-based clustering method (Reynolds, 2008; Belton et al. 2013). A GMM describes the distribution of points in a feature space. Each of the generated Gaussian models represents a local cluster in the feature space. However, distribution-based algorithms put an extra burden on the user: concisely defined mathematical models that are not available for many. In density-based clustering, clusters are defined as point groups with higher density. Objects in sparse areas - that are required to separate clusters - are usually considered to be noise or border points.

The most popular density-based clustering method is density-based spatial clustering of applications with noise DBSCAN (Ester, et al., 1996). It connects points within a certain distance-threshold; it only connects points that satisfy a density criterion. Clusters are defined according to the local-density pattern of points and based on the fact that each cluster is surrounded by an area of lower density (or even equal to zero) representing the noise. The DBSCAN algorithm requires two parameters – searching radius (ε) specifying the maximum distance between two points for them to be considered in a cluster, and a minimum number of points (MinPts) specifying how many neighbors a point should have for them to be considered as a cluster. The algorithm starts with an arbitrary starting point that has not been "visited" and retrieves this point's ε-neighborhood. If it contains sufficient points, a cluster is started; otherwise, the point is labeled as noise. The point that does not meet the criteria might later be found in a sufficiently sized ε-environment of a neighbor point and hence be made part of a cluster. If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of the same cluster. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed to discover a new cluster or noise. The testing results indicated that an appropriate selection of MinPts and ε could improve the

accuracy of object clustering. Because DBSCAN provides the required accuracy and efficiency without asking predefined number of objects, it was selected as the base of object clustering in this project.

2.3 Vehicle and Pedestrian Classification

In the past decade, researchers have used LiDAR and vision-based approaches for pedestrian and vehicle detection and tracking (Premebida et al., 2007; Premebida et al., 2009; Ismail et al., 2009; Sivaraman and Trivedi, 2013). Numerous algorithms were developed to improve detection accuracy and reduce computational expense for reliable onboard sensing systems. For onboard systems, the major approach to identifying pedestrians and vehicles is feature-based machine learning classification (Wojke and Haselich, 2012; Azim and Aycard, 2012; Cheng et al., 2014). Existing feature-based methods/algorithms heavily depend on the high density of laser points that provide detailed descriptions or specific characteristics of the objects. Due to the required detection distance and the low number of laser beams, roadside LiDAR sensors can only provide limited and sparse laser points of objects on the road. The detection accuracy of feature-based machine learning algorithms was observed to be significantly low when being tested with roadside LiDAR data in this project. In addition, most existing onboard-sensing systems (Premebida et al., 2007; Himmelsbach et al., 2008; Premebida et al., 2009) combine data from multiple sensors such as LiDAR and cameras, but the roadside LiDAR system relies on LiDAR sensors only in the current deployment that is limited by edge-computer configuration, mainteNACe resource, and space for additional cameras. Support vector machine (SVM) (Kidono, et al., 2011), neural network (NN) (Szarvas, et al., 2006), and Naive Bayes (NB) (Spinello and Siegwart, 2008) were also used by onboard sensing systems for vehicle and pedestrian classification. The computational demand of NN classifier is lower than that of SVM and NB, so NN was selected for object identification with roadside LiDAR data.

2.5 Object Tracking

Tracking is to associate clusters representing the same vehicle/pedestrian of different frames. Common methods used for object tracking in LiDAR data include nearest neighbor (NN), Kalman filter (KF), multiple hypothesis tracking (MHT), and other algorithms. Azim and Aycard (2012) built object association by finding the nearest clusters of continuous frames, so their method was called Global Nearest Neighbor (GNN). While finding a possible nearest neighbor association between a track and an observation, the algorithm ignores associations in different classes. For instance, if the nearest neighbor of a vehicle observation classified as a pedestrian, the algorithm ignores this nearest neighbor and searches for the next nearest point within the specified range. Miyasaka et al. (2009) used an extended Kalman filter (EKF) to estimate the non-linear states of objects and to predict their locations in the next frame. Himmelsbach et al. (2008) combined the Kalman Filtering and nearest neighbor for vehicle tracking. Mortion and Underwood (2011) developed a dynamic object tracking algorithm that uses a motion model to predict the likely location of future observations then associates these observations with the current set of tracks. Multiple hypothesis tracking (MHT) algorithm effectively mitigates association errors by retaining multiple data association hypotheses until enough evidence is accumulated to resolve past assignment ambiguities. Variations on both the

implementation and the ordering of these steps resulted in different tracking methods. Most of the methods mentioned above were developed for processing on-board LiDAR data.

2.6 Literature Review Conclusion

Results of the literature review showed that there is a lack of data-processing algorithms for a complete procedure although a great amount of research has been performed for processing onboard LiDAR data. It is necessary for this project to develop the required methods and procedure, so roadside LiDAR data can be processed accurately and efficiently to extract multimodal traffic trajectories. This project will provide the foundation for data-processing methods to apply roadside LiDAR sensors in various traffic scenarios.

ROADSIDE LIDAR DATA PROCESSING

3.1 LiDAR Sensor and Data

Velodyne VLP-16 LiDAR sensors were used in this project to collect roadside LiDAR data and test deployment methods. The sensor unit is composed of 16 laser/detector pairs and an inner motor that rotates lasers horizontally to generate a 360-degree three-dimensional (3D) point cloud (600,000 points per second) in a 100-meter radius. The rotation frequency can be customized from 5 Hz to 20 Hz. One data frame is generated after the sensor finishes a 360-degree scan and is stored in the packet capture (pcap) format. The sensor covers a 30-degree vertical field with the 16 laser beams evenly distributed in the range of 15-degree up and 15-degree down, which means a 2-degree interval between adjacent laser beams. The VLP-16 sensor reports spherical coordinates (r, ω, α) that can be converted into Cartesian coordinates (x, y, z). The LiDAR data include point location (x, y, z), intensity, laser ID, azimuth, the distance between a data point and the sensor, adjusted time, and timestamp (Velodyne, 2016). By knowing LiDAR coordinates and GPS locations of the LiDAR sensor and a reference point, all LiDAR points can be projected to a geography coordinate expressed by longitude, latitude, and elevation. A LiDAR sensor can be temporarily installed on a portable platform (like a tripod) for short-term data collection or permanently installed on roadside infrastructure for long-term data collection and monitoring. This study showed that 6-7 ft height is optimal for roadside LiDAR deployment, which provides expected detection range and capability of overcoming occlusion caused by passenger vehicles because the sensor location is higher than passenger cars. The data can be viewed in the open-source software VeloView (https://www.paraview.org/veloview/). Velodyne was in the process of manufacturing new 16-channel LiDAR with a narrower interval when the project team was preparing this report, so the new sensors were not tested in this project. It is important to test the new sensors and validate the influence of the changed sensor intervals in future research.

3.2 Background Filtering

An automatic background filtering method named 3D-density-statistic-filtering (3D-DSF) was developed in this project to filter both static (buildings) and dynamic background objects (shaking trees and bushes) in roadside LiDAR data. The 3D-DSF considers the aggregated-density distribution of LiDAR points in the 3D space. The 3D-DSF method includes four major steps: frame aggregation, point density statistics, threshold (TD) learning, and real-time filtering. A 3D matrix representing the 3D sensing space is generated in the third step, and each cell is labeled as background or non-background based on its number of points. In real-time background filtering, the LiDAR points in the background cells are excluded from LiDAR data before being processed in the following steps. It can be used to process roadside-LiDAR data regardless of brand or number/angle of laser beams. Figure 3-1 shows the 3D-DSF flowchart.

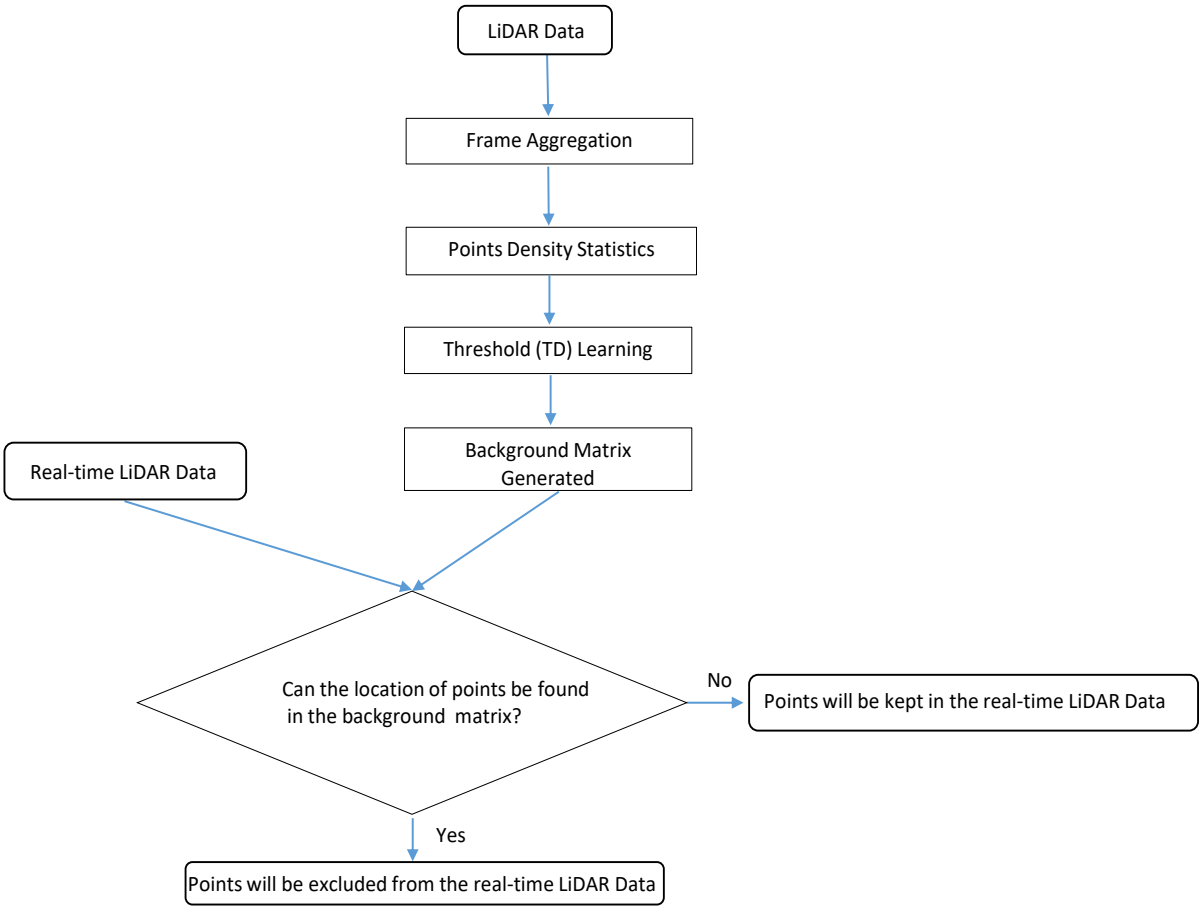


Figure 0-5 Flow chart of the developed 3D-DSF algorithm

Frame Aggregation

3D-DSF aggregates LiDAR data frames collected in a period as initial input for background learning. Theoretically, the more frames used for background identification, the better accuracy can be achieved. However, more frames require more computer memory and increase the time for background identification. In contrast, the algorithm is unable to identify vibrating background points accurately with a low number of aggregated frames. This research compared the performance of different amounts of frames to determine the optimum number for aggregation. When the number of aggregated frames was higher than 3000, increased frames did not significantly improve the accuracy of background identification. Therefore, the recommended number of aggregated frames is 3000.

Points Statistics

After frame aggregation, the algorithm divides the LiDAR sensing space into continuous 3D cubes. Each cube is a background space or non-background space. A 3D matrix is built to represent the whole space so that each matrix element represents a cube and its value is the number of aggregated points in it. A critical parameter of this algorithm step is the cube side length that determines the resolution of background identification and size of the 3D matrix. A short side length provides higher resolution but can significantly increase the size of the 3D matrix and requirement of computer memory; long side length requires lower computer memory but may report non-background points as background because of the reduced

resolution. Influence of side length on the performance of the algorithm was explored in this project. Time for generating the 3D matrix also changes along with the selected side length. When the side length was changed from 0.1m to 0.05m, the accuracy of background filtering was not improved significantly but the time cost increased by almost 40 minutes. Therefore, the recommended cube-side length is 0.1m.

Threshold Learning

The algorithm calculates the density of aggregated 3D points in each cube in this step. In general, the density of a cube with background should be higher than the density of a cube in traffic lanes with moving vehicles or pedestrians. A threshold of cube density (TD) is needed to distinguish background and non-background cubes. A low TD may misidentify roadway space as background because of slow-moving or high-volume vehicles or pedestrians; a high TD may not correctly identify the background cubes that are far away from the LiDAR sensor with relative low point density. The point density of the same object scanned by the same LiDAR varies with its distance from the sensor. In general, the density is lower when the object is further away from the sensor. Figure 3-2 shows an example of the point density change of a vehicle when it is at different distances from a sensor.

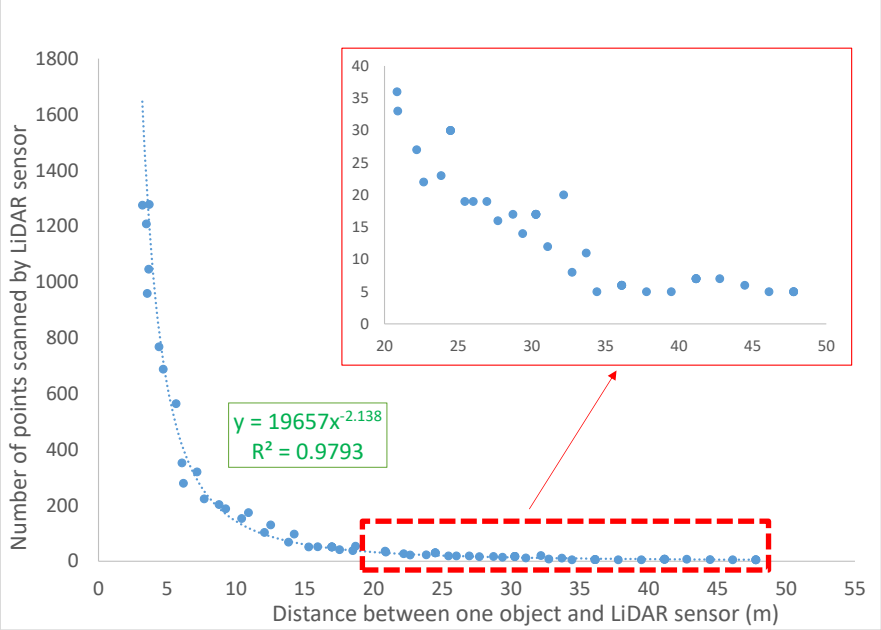


Figure 3-0-6 Distribution of vehicle points in LiDAR data

When background objects are at different distances from a LiDAR sensor, the point densities in background cubes are also different, which means the TD should vary in different detection ranges. The algorithm divides the whole sensing range into six areas, based on experience, for background identification and filtering: 0-5m, 5-10m, 10-20m, 20-30m, 30-40m, and longer than 40m from the LiDAR sensor. The TD threshold is determined for each subarea by Equation 1:

$$Slope = \frac{F_i - F_{i-1}}{N_i - N_{i-1}} \quad (1)$$

where *Ni* is the *i*th number of points per cubic (from lowest to highest) and

F is the frequency of the *i*th number of points per cube.

When the slope firstly becomes 0 or positive, the frequency of number of points per cube in Equation 1 (*F*) is used as TD. After identifying the background cubes with the determined thresholds, the background space is labeled in the 3D matrix. For real-time data processing, any LiDAR point in the background cubes is identified as a background point and excluded from the following process.

Background Filtering Case Study

For testing the background-filtering algorithm, it was applied to process roadside LiDAR data collected at six sites in Reno, Nevada, as summarized in Table 3-1.

Table 0-2 Sites for Background Filtering Evaluation

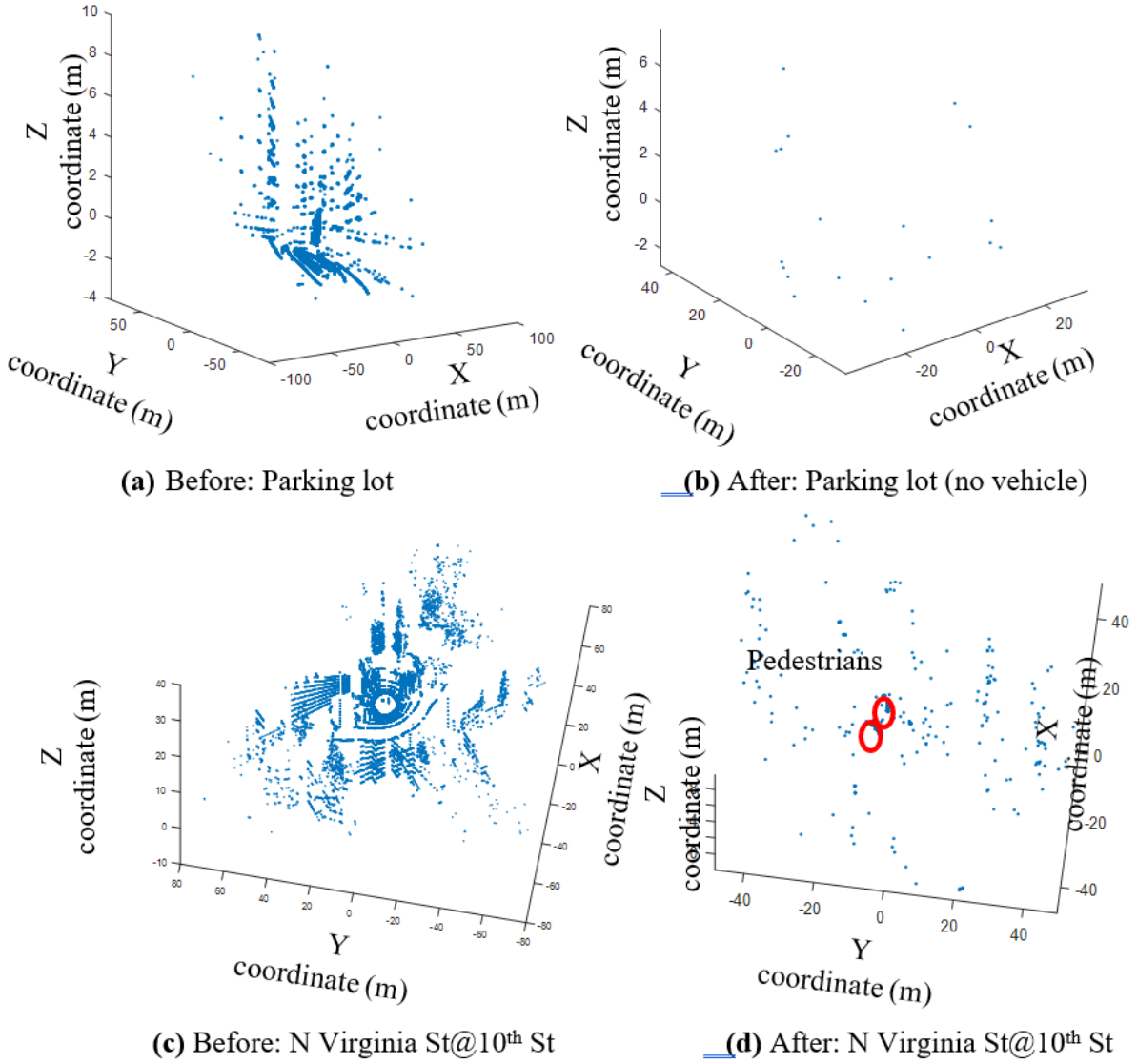
Location	Frames aggregated for background identification	Sensor rotating frequency (Hz)	AADT	Speed limit (mph)	Time of Data Collection
N Virginia St. at15th St	3000	10	11000	25	Night time: no pedestrians or vehicles
N Virginia St. at10th St	1500	5	11000	25	Peak hour: high pedestrian and vehicle volume
Evans Ave at Enterprise Rd	2500	10	3500	25	Non-peak hour
Parking Lot	3000	10		5	Non-peak hour
Kietzke Lane	3000	10	22000	45	Peak hour: high vehicle volume
I80 in Elko, NV	3500	10	11800	80	Non-peak hour

A LiDAR sensor was permanently installed on top of a pedestrian-signal head at the intersection of N. Virginia St. and 15th St., as shown in Figure 3-3(a). Data at other sites were collected using a VLP-16 LiDAR mounted on a tripod for short-term data collection, as shown in Figure 3-3(b).

(a) Permanent installation (b) Temporary installation

Figure 0-7 Different installation methods of roadside LiDAR sensors

Two examples of frames before and after background-filtering are presented in Figure 3-4 with the raw data collected at the parking lot and the intersection of N. Virginia St. and10th St.



(a) Before: Parking lot

(b) After: Parking lot (no vehicle)

(c) Before: N Virginia St@10th St

(d) After: N Virginia St@10th St

Figure 0-8 Examples of before and after background-filtering

Background space cubes were accurately identified, and background points were successfully removed when there was no traffic (Figure 3-5 (a) and (b)) in aggregated frames. The noise points left in Figure 3-4 (b) were not background objects but random noise in LiDAR data. When many vehicles and pedestrians existed during data collection, some low-density background points were left after background-filtering (Figure 3-5 (c) and (d)). The left background points and noise points discretely distributed in the sensing space, so they were further filtered in the following step – clustering. However, it is still recommended to use LiDAR data frames with low/no traffic for background identification. Table 3-2 presents the evaluation results of filtering background points from data collected at the various sites.

As shown in Table 3-2, the developed method filtered more than 99% of background points when there were no pedestrians or vehicles (N. Virginia St. at15th St.). Even at the site with high pedestrian and vehicle volume (N. Virginia St. at 10th St.), the method still removed more than 97% of background points. For the data frames with pedestrians and vehicles, the

background-filtering process needs to keep LiDAR points of pedestrian and vehicle as much as possible so that shapes of point clusters are not changed. In this evaluation, the vehicle length was calculated by manually selecting the front point and back point of each vehicle in data frames before and after background filtering. The results in Table 3-2 show that vehicle shapes were only minorly changed after background filtering because of loss of sparse edge points. The sites included traffic environments with different function classes, geometries, and speed limits (from 5mph to 80mph). The results show that the 3D-DSF can effectively filter background points at all those sites. Though there were noise points left after the background filtering, the following data-clustering step, using k-means clustering or density-based clustering methods, was not influenced.

Table 0-3 Evaluation of Background-Filtering Performance at Different Sites

		UNR Parking lot	N .Virginia St. at 10th St.	Evans Ave at Enterprise Rd.	N. Virginia St. at 15th St.	Kietzke Lane	I80 in Elko, NV
Before	Background & noise points	18316	45132	18621	18035	18217	17162
After	Background and noise points	21	1362	176	25	46	20
Filtering Percentage		99.8%	97.0%	99.1%	99.9%	99.7%	99.9%
Before	Vehicle points	NA	507	451	NA	531	651
After	Vehicle points	NA	493	411	NA	506	606
Vehicle-point lost percentage		NA	2.8%	8.9%	NA	4.7%	6.9%
Before	Detected Vehicle Length (ft)	NA	46.1	44.8	NA	43.2	51.6
After	Detected Vehicle Length(ft)	NA	40.1	42.5	NA	40.0	49.1
Before	Pedestrian points	NA	209	553	NA	NA	NA
After	Pedestrian points	NA	196	532	NA	NA	NA
Pedestrian-point lost percentage		NA	6.2%	3.8%	NA	NA	NA

Note: the detected vehicle length is not exactly the real vehicle length subject to the vehicle location.

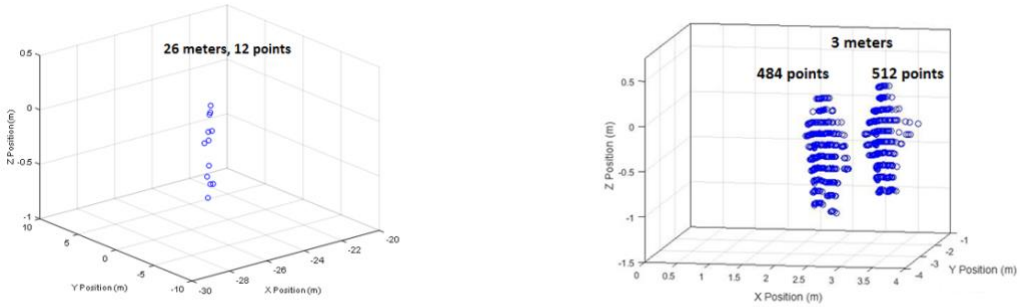
3.3 Detection and Tracking of Pedestrians and Vehicles

Background filtering is followed by object identification that contains two sub-steps: clustering and classification. Clustering groups of LiDAR points belonging to a same object and classification is used to identify a road user's type – vehicle, pedestrian or others. This project developed a clustering algorithm based on the traditional DBSCAN method (Ester et al., 1996) with consideration of spatial distribution and density difference of LiDAR points. A new vehicle-pedestrian-classification method was also developed for the roadside-LiDAR sensing system to identify the type of each LiDAR point cluster.

3.3.1 Objects Clustering

DBSCAN is one of the most popular density-based clustering algorithms (Ester et al., 1996; Aggarwal and Reddy, 2013). Two parameters of the traditional DBSCAN algorithm are Minimal Points (MinPts) and the searching radius (ε) that determine the performance of clustering. If the number of data points within a searching area (the predefined searching radius) is greater than or equal to the predefined MinPts value, those data points are clustered to form a group. By the end of DBSCAN, laser points are divided into three categories – core points, border points, and noise points. Core points and border points are grouped to describe the shape of each object. An advantage of the DBSCAN method is that it does not need a predefined number of clusters. This is especially useful when clustering road user points because the number of objects on roads is unknown before clustering.

Due to the angles of LiDAR laser beams and shape/size of pedestrians and vehicles, the number of LiDAR points reflected by a pedestrian is different from the number of points from a vehicle at the same distance from the sensor. The number of LiDAR points from the same object also changes at different distances from a LiDAR sensor. As an example, Figure 3-5 (a) and Figure 3-5 (b) show 3D points of pedestrians at different distances from a VLP-16 LiDAR sensor. There were only 12 points from a pedestrian standing 26 meters away from the sensor, while 484 and 512 points are available for two pedestrians three meters away from the same sensor.



(a) Pedestrian data points at 26 meters from LiDAR. **(b)** Pedestrian data points at 3 meters from LiDAR.

Figure 0-9 Comparison of obtained points from pedestrians at different distances from a VLP-16 sensor

Therefore, it is difficult for the traditional DBSCAN algorithm to cluster roadside-LiDAR points accurately with a fixed MinPts value and a fixed searching radius. The MinPts value and searching radius need to be adjusted at different distances from the sensor, so a

DBSCAN-based clustering algorithm was developed to implement adaptive MinPts values and searching radiuses. When the MinPts values and searching radiuses (vertical and horizontal) change at different distances from a sensor, the calculation for each LiDAR point leads to a heavy computation requirement. Therefore, the proposed procedure divides the LiDAR detection range into three sub-areas (I, II, III) based on the distances from the sensor (as shown in Figure 3-6). Each subarea uses the same searching radius and MinPts values determined by the outer edge (farthest to the sensor) of the subarea. The developed clustering method uses different radiuses in the vertical direction (longer radius) and the horizontal plane (shorter radius), and generates an ellipsoid searching space rather than a sphere. The vertical and horizontal searching radiuses were determined by the vertical LiDAR point resolution (determined by the vertical angle of adjacent laser beams) and the horizontal LiDAR point resolution (determined by the shooting frequency of a laser beam). Therefore, the new method can separate points of two pedestrians close to each other. For example, the vertical distance between two adjacent laser beams needs to be shorter than the height of a road user to avoid the object falling between two laser beams and not being detected. The vertical angle between two adjacent laser channels of VLP-16 sensors is 2 degrees and the vertical height between two adjacent laser beams is about 1 meter at a 30-meter distance from the sensor, so a pedestrian higher than 1 meter can be detected (scanned by at least one laser beam) in this range.

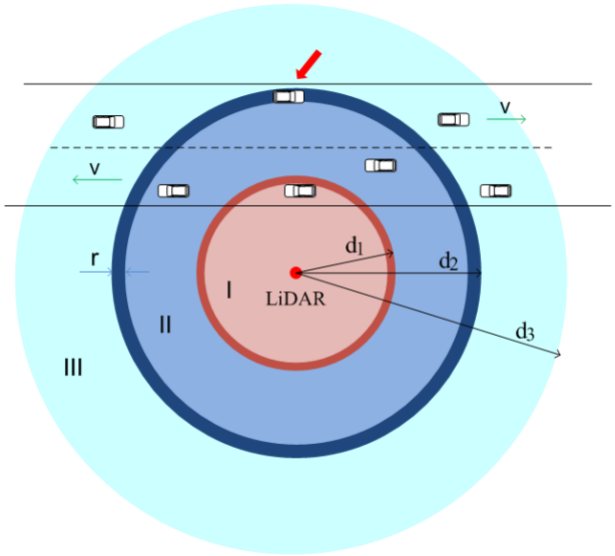


Figure 0-10 Clustering area division for using different parameters

3.3.2 Pedestrian and Vehicle Classification

It is critical to distinguish pedestrians and vehicles in point clusters. The classification method developed in this project extracts three features from point clusters as listed in the following:

1) Total number of points

A cluster is a maximal set of density-connected points. Although occlusion could affect the total number of points in clusters, in general, a vehicle cluster includes more points than a pedestrian cluster when they are at the same distance to the sensor.

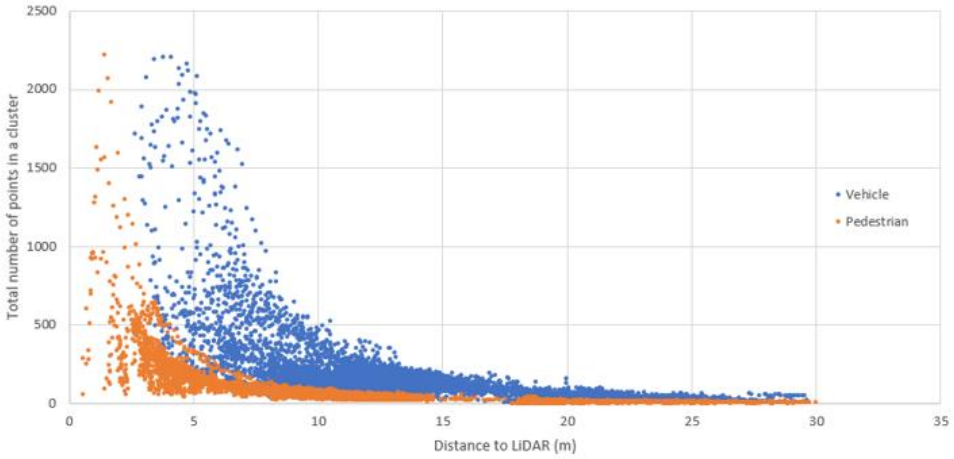
2) Cluster-sensor distance

The distance to the LiDAR sensor influences the number of points in a cluster. The mean of all point-sensor-distances in a cluster is used as the cluster-sensor distance.

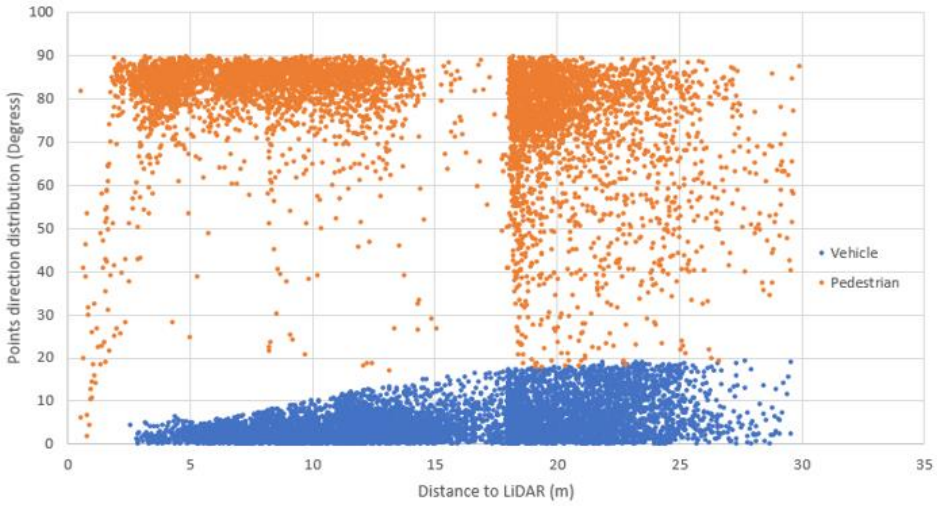
3) Point-distribution direction

Space distribution of points belonging to a pedestrian cluster is mainly along the vertical direction (z-axis), while the distribution of points in a vehicle cluster is primarily along the horizontal direction (parallel to the x-y plane) in general. With the least-square linear regression method, a linear function can be generated to describe the point-distribution direction of each cluster.

Figure 3-7 shows the difference between the three extracted features between vehicles and pedestrians. In Figure 3-7 (a), the number of points in a cluster decreases along with the increase of cluster-sensor distance, and a vehicle cluster contains more points than a pedestrian cluster at the same cluster-sensor distance. The difference between point-distribution directions between pedestrian clusters and vehicle clusters is shown in Figure 3-7 (b). When the angle between a cluster's point-distribution direction and the x-y plane is less than 20 degrees, this cluster has a high possibility of being a vehicle. It needs to be noted that the first two features, number of points and distance, are related. The distribution direction does not have a direct connection with the number of points and distance. These three features were used as inputs of object classification.



(a) Cluster-sensor distance and number of cluster points.



(b) Cluster-sensor distance and point-distribution direction.

Figure 0-11 Difference of classification features between vehicles and pedestrians

In this project, a classification model based on the backpropagation artificial neural network (BP-ANN) was developed to distinguish pedestrians and vehicles in clusters. The BP-ANN (as in Figure 3-8) is a multilayer feed-forward neural network composed of an input layer, a hidden layer (or layers), an output layer, and neurons in each layer. The number of hidden layers can be more than one. The input data is fed to the input layer; inputs and weights between nodes in different layers determine the output. The BP-ANN model can be trained by adjusting the weight values between nodes with a dataset that has known input (the three cluster features in this project) and output (road user type in this project) information. When the minimal error is reached, or the number of iterations is beyond the predefined value, the training process ends and provides a trained neural network that can estimate the cluster type of the new input. It is found that the selected features are critical for the performance of traveler type classification, so the simple BP-ANN classification method can provide high accuracy with the features showing a clear difference between different types. In contrast, an advanced classification method is not necessary to provide better accuracy if selected features are not appropriate.

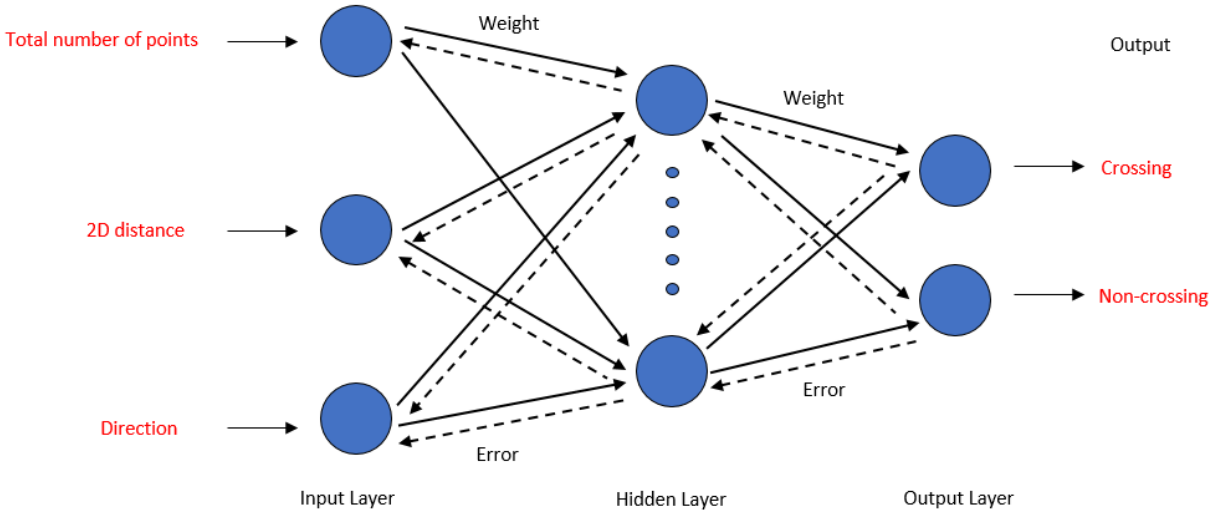
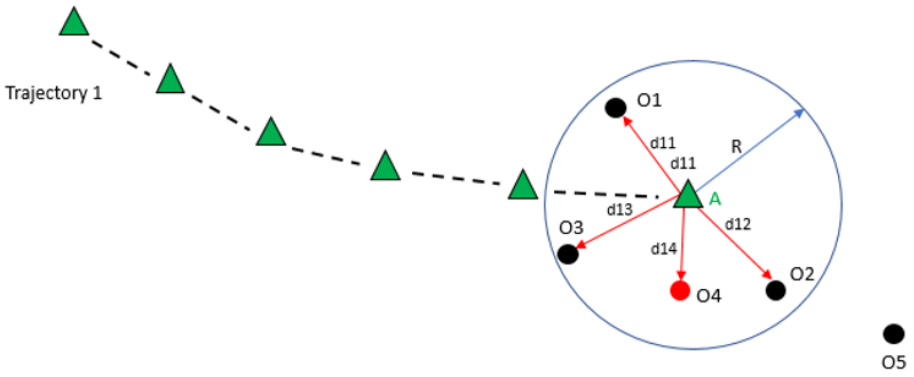


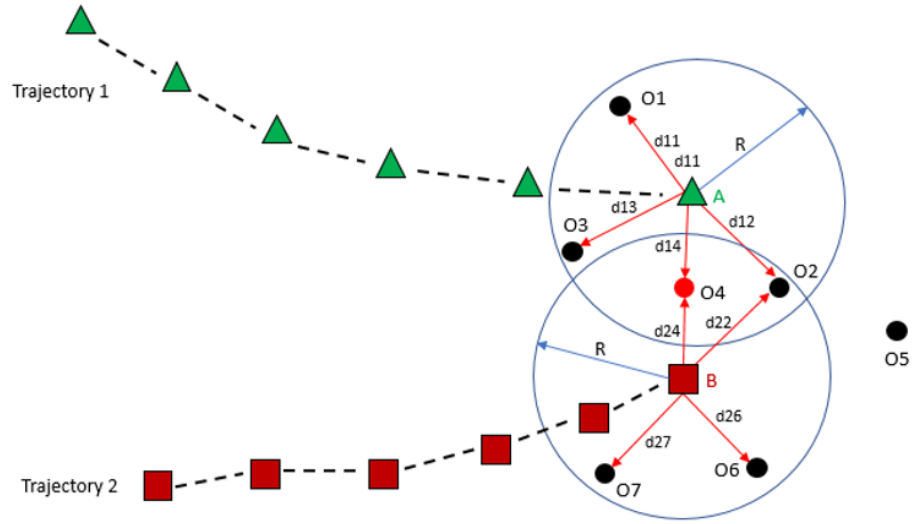
Figure 0-12 Structure of BP-ANN model

3.3.3 Tracking

In order to obtain each road user's trajectory and speed, object tracking was performed after clustering and classification. Tracking identifies the same object in continuous data frames (Coifman et al., 1998; Allodi et al., 2016; Wang et al., 2017). Two factors are considered for object association: distances between an object in a previous frame to all objects in the current frame and the time difference between two considered frames. An object in the current frame is matched to an object in the previous frame if the distance between them is the shortest among all candidate objects and time difference between the two data frames is within a certain period. The candidate objects are selected by the area within a distance threshold that is determined by the historical object speed and the time difference between frames. Each tracked object uses the same object ID in different frames. When an object in the current frame cannot find a matched object in its previous frame, a new tracking ID is assigned. Figure 3-9 shows two cases of tracking. In Figure 3-9 (a), objects O1-O4 in the searching region (radius R was determined by the historical speed) are considered as candidates. Among the four candidate objects, O4 is the nearest object to object A, so O4 is the updated location of object A. In Figure 3-9 (b), O4 is the searching area of object A and object B. In this case, O4 is matched to the object with the shortest distance (min{d_{14} and d_{24}}) and used as that object's updated location. For some frames where clusters cannot be detected (occlusion or failure of clustering), the Kalman filter can be used to predict the status of the missing object, thus improving the tracking continuity. After successful object tracking, a discrete Kalman filter model is updated with the position/speed/acceleration information of the object in the previous frame and the position/speed/acceleration information of the matched object in the current frame.



(a) Single trajectory tracking.



(b) Multiple trajectories tracking.

Figure 0-13 Demonstration of object tracking

With each road user's trajectory identified, position, velocity, and direction can be determined. Velocity is the distance between the same object's locations in two frames divided by the time difference. When a vehicle is approaching the LiDAR, the position reference point is the front corner point. When a vehicle is leaving the LiDAR, the position reference point is the back corner point. An analysis was performed to justify that the vehicle travel distance between adjacent frames (at 10 Hz) is much shorter than the distance between adjacent vehicles in a road lane, so object association based on the shortest distance works in this process. Figure 3-10 presents a comparison of a vehicle's frame-to-frame travel distance and the distance between different vehicles in the same lane. The curves of 1-second headway, 2-second headway, and 3-second headway describe the closest vehicle distance in one lane with the various headways at different speeds. The curve of frame-to-frame distance at 10 Hz presents the travel distance of the same vehicle in adjacent frames (recorded every 0.1 second) at different speeds. The comparison shows that the vehicle travel distance between adjacent frames is much shorter than the distance between vehicles in the same lane, so the shortest distance method will connect the same object's clusters in adjacent frames rather than mismatching different objects' clusters.

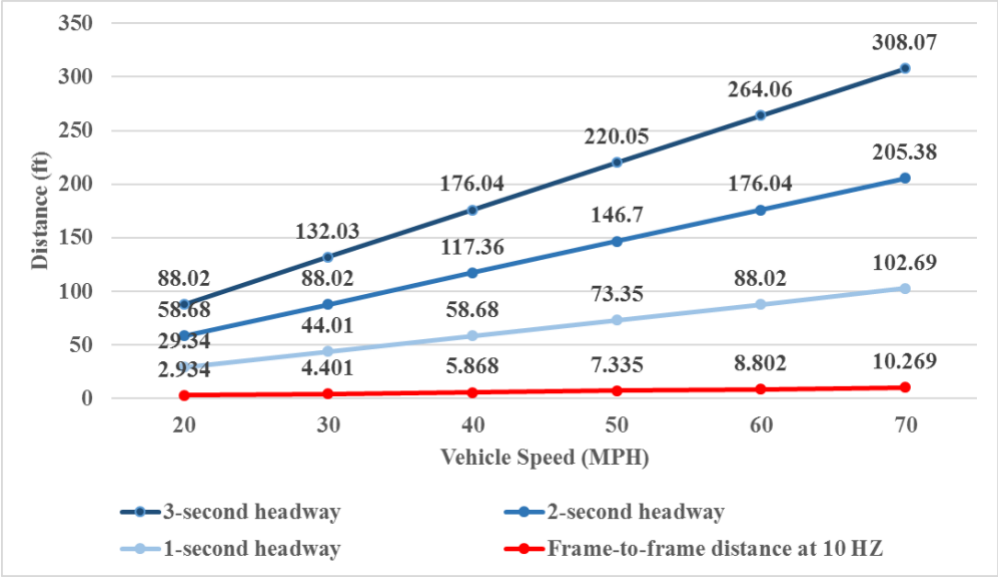


Figure 0-14 Comparison of frame-to-frame travel distance and distances between adjacent vehicles in the same lane.

3.4 Validation of Detection and Tracking Methods

Three field tests were conducted at three intersections (shown in Figure 3-11) in Reno, Nevada to evaluate the proposed procedure and methods for detecting and tracking road users with a roadside LiDAR sensor. The intersection of North Virginia Street and 15th Street (Site 1) is a signalized intersection with permanently installed LiDAR sensors. Site 2 was the intersection of Sierra Street and 11th Street that is a two-way-stop-sign controlled intersection with Rectangular Rapid Flash Beacon (RRFB) pedestrian signals. A 16-channel LiDAR sensor with 10Hz rotation frequency was mounted on a tripod (approximately 6ft above the ground) for data collection. Site 3 was the T intersection of North Virginia Street and 10th Street where the LiDAR sensor was also installed on a tripod (approximate 6ft height above the ground) with 5Hz frequency. Speed limits of the three sites were 25 mph. Table 3-3 summarizes the data used in this validation from the three sites.

Table 0-4 Data collection site information

Location	LiDAR rotation frequency (Hz)	Data collection period	Speed limit (mph)
N. Virginia St. at 15th St. (Site 1)	10	19860 frames (33min)	25
Sierra St. at 11th St. (Site 2)	10	18000 frames (30min)	25
N. Virginia St. at 10th St. (Site 3)	5	18000 frames (60min)	25

Figure 0-15 Google map of data collection sites

For the clustering process, all road users were clustered by the modified DBSCAN algorithm. An example of clustered points in a data frame is shown in Figure 3-12 with three vehicles and three pedestrians. Two pedestrians (represented by red and black dots) with a distance of 0.5 meter were clustered accurately. Table 3-4 presents evaluation results that compare the number of extracted clusters from LiDAR data and identified road users from recorded videos. The detection accuracy of object clustering was about 96%. The main reason for clustering failure was that only sparse points were collected from objects at a far distance, and occlusion is another issue.

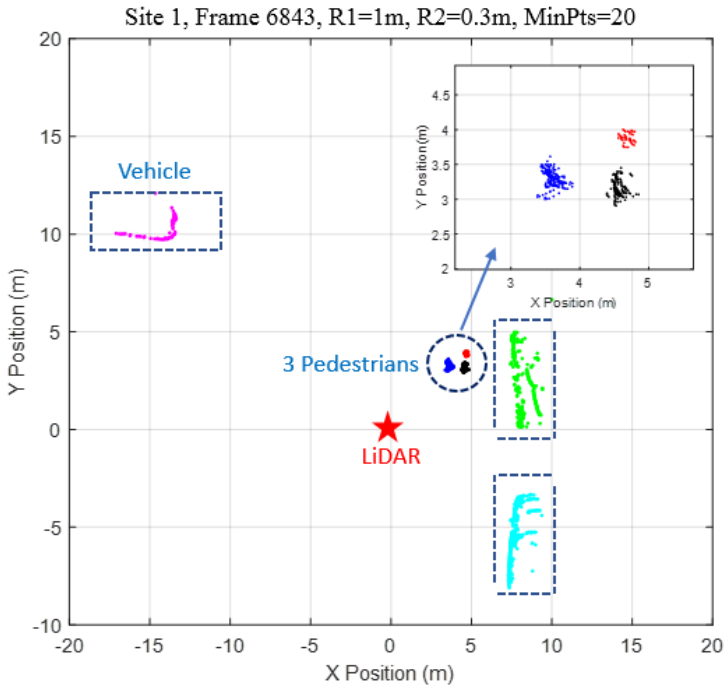


Figure 0-16 Demonstration of clustering results

Table 0-5 Summary of clustering evaluation results

Location	Total frames	Clusters from the proposed algorithms	Objects from the video	Detection Accuracy
N. Virginia St. at 15th St.	100	315	305	96.8%
Sierra St. at 11th St.	100	356	338	95.0%
N. Virginia St. at10th St.	100	243	234	96.2%

After clustering, the next step is to classify clusters into vehicles and pedestrians using the BP-ANN model and the three extracted cluster features: number of points, distance to the sensor and point-distribution direction. LiDAR data were recorded and reviewed to train the ANN model. The training dataset was further divided into three categories for Training (70%), Validation (15%), and Testing (15%) steps. An evaluation dataset that has not been seen by the trained model was used to evaluate the performance of the trained network. Different training clusters might be the same pedestrian or vehicle in different data frames. With the training data, the project team first determined the number of neurons in the hidden layer. Because there is no common approach to determining the number of neurons, ten neurons are selected. Then, the number of neurons can be changed and evaluated to identify the optimum one. After the determination of the number of neurons, the research team tested various training functions and transfer functions. It was found that the performance of the ANN model was not obviously influenced by the difference of training functions and transfer functions. Scaled conjugate gradient backpropagation for training function, softmax transfer function and mean squared error performance function were selected as functions of the ANN model. The classification

results are listed in Table 3-5. An approximate 96% classification accuracy can be guaranteed within the 30-meter-radius sensing range. The classification failure was found to be caused by vehicle occlusion.

Table 0-6 Evaluation of cluster classification

Locations		Observed pedestrians and vehicles		Recognized clusters		
	Total frames	Pedestrian clusters	Vehicle clusters	Pedestrian clusters	Vehicle clusters	Classification rate
N. Virginia St. at 15th St.	1000	403	597	385	581	96.6%
Sierra St. at 11th St.	1000	389	611	370	593	96.3%
N. Virginia St. at 10th St.	1000	284	716	275	690	96.5%

For the tracking procedure, a total of 1,023 vehicles and 48 pedestrians were detected and tracked from 18000-frame data collected at Site 1. Figure 3-13 presents part of the extracted vehicle and pedestrian trajectories. The red and blue dots represent trajectories of vehicles and pedestrians at 10 Hz frequency. Table 3-6 shows the tracking results from 1000 data sample frames at three sites. Tracking accuracy of approximately 95% can be achieved within about 30 meters detection range from the LiDAR sensor. Similar reasons have been found for tracking failure as those addressed in the clustering section.

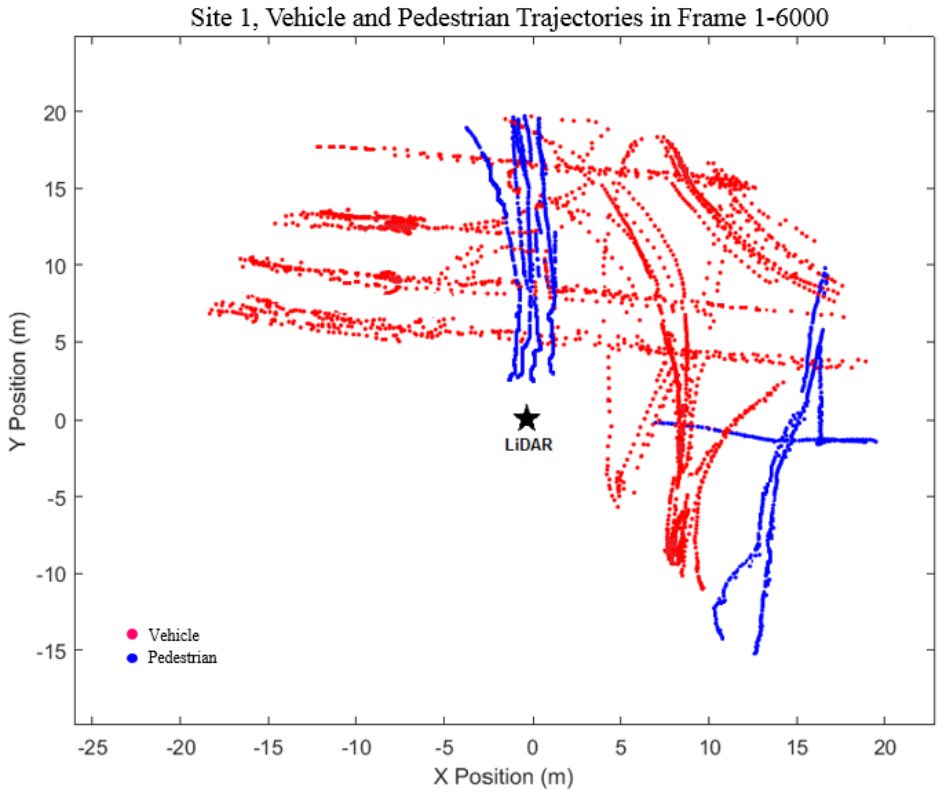
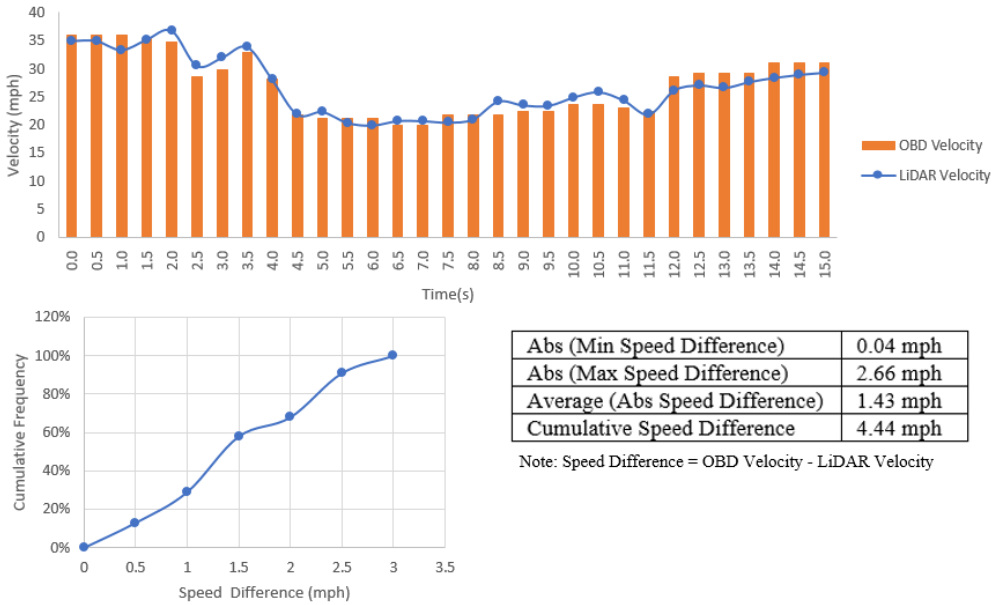


Figure 0-17 Example of extracted vehicle and pedestrian trajectories

Table 0-7 Evaluation of object tracking

Location	Total frames	Trajectories from the proposed algorithm	Trajectories from the video	Tracking accuracy
N Virginia St@15th	1000	40	42	95.2%
Sierra St@11th St	1000	35	37	94.6%
N Virginia St@10th St	1000	34	35	97.1%

Evaluation of detection/tracking performance also included speed validation that compares speed extracted from roadside LiDAR data and speed logged through the OBD-II vehicle interface. The OBD-II speed was from the vehicle speedometer and was considered as accurate speed. Vehicle speed was calculated every 0.1 second (10 HZ) using the LiDAR data, while the OBD logger provided the speed every 0.5 seconds. So, the validation process was to compare the space mean speed from the LiDAR sensor (in every 0.5 seconds) with speed measured by the onboard system. Figure 3-14 shows the statistical testing results of speed validation. The orange bars show speeds obtained from the speedometer of a testing car running between 25 to 35 mph; the blue points depict the calculated speeds from LiDAR data. The statistic results show that 90% of the speed difference is lower than 2.5mph. The minor speed difference was caused by the timestamp and point position offsets between frames.

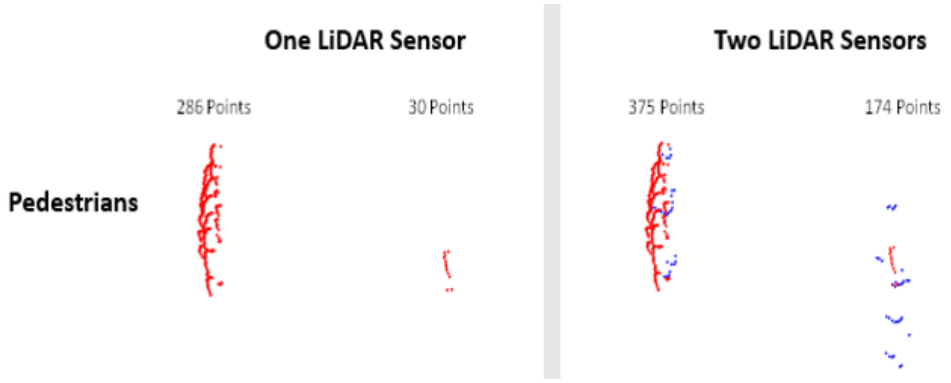


Abs (Min Speed Difference)	0.04 mph
Abs (Max Speed Difference)	2.66 mph
Average (Abs Speed Difference)	1.43 mph
Cumulative Speed Difference	4.44 mph

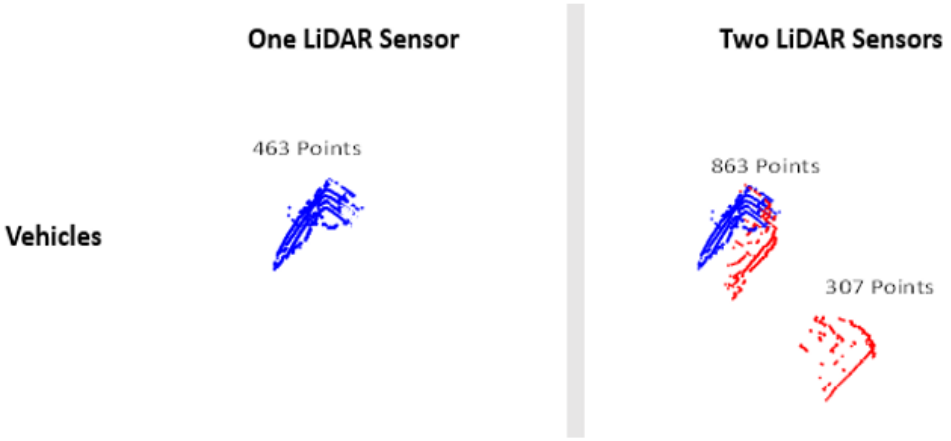
Note: Speed Difference = OBD Velocity - LiDAR Velocity

Figure 0-18 Validation of speed information extracted from roadside LiDAR

Occlusion is the major challenge of LiDAR-point clustering, classification and object tracking. The LiDAR location is suggested to be 7-ft high from the ground, so the sensor is higher than most light-weight vehicles (passenger cars) and can at least scan the top of vehicles if they are blocked by passenger cars. Complete occlusion is normally caused by trucks and buses. A recommended solution is to install multiple LiDAR sensors at different corners of intersections and along both sides of roads, thus providing detection coverage from various sensors in different directions. Figure 3-15 shows an example of partial and full occlusion cases. After integration LiDAR points from two LiDAR sensors, the second pedestrians in Figure 3-15 (a) can be identified with more points and better shape information. In Figure 3-15(b), the second vehicle was totally blocked by the first vehicle when using one LiDAR sensor, but it can be detected by the second LiDAR sensor at a different location. Therefore, the LiDAR sensor network will extend the detection range and reduce the possibility of occlusion.



(a) Partial occlusion.



(b) Full occlusion.

Figure 0-19 Demonstration of multiple-LiDAR overcoming occlusion

3.5 LiDAR Data in Inclement Weather

This project also tested the performance of roadside LiDAR sensors and the developed data processing methods in inclement weather conditions. This test applied a dataset of 27,000 frames under different weather conditions (9000 frames in good weather, 9000 frames in rainy weather, 9000 frames in snowy weather). The results were compared and illustrated in Table 3-5. The results showed that during good weather, the cluster method provided a low false rate of 0.14%. Those vehicles that were not successfully detected were mainly caused by vehicle occlusion, which can be solved by deploying multiple LiDAR sensors on both sides of the road. The average detection range in good weather was 38 meters. In rainy weather, the false rate went a litter higher than that in good weather, but it was still in a low range. When it was snowing, the clustering accuracy was 98.13%. There were 35 false vehicle detections caused by snowflakes. The average detection range was 35.4m, which was a little shorter than that under good weather and rainy weather. The false vehicles could be excluded by comparing the identified vehicles in continuous frames, which was not included in this study. It needs to be noted that data used for the evaluation of weather influence was different from the dataset used in Section 3.4. The clustering accuracy was higher than the accuracy documented in Section 3.4 because the traffic volume in the weather influence evaluation is lower and fewer occlusion

events occurred.

Table 0-8 Evaluation of Roadside LiDAR and Developed Data Processing Methods in Inclement Weather

	Good Weather	Rainy Weather	Snowy Weather
The actual number of vehicles extracted from LiDAR videos (AV)	35,405	23,642	5,413
Detected number of vehicles (DV)	35,354	23,410	5,382
Number of false vehicles created by snowflakes/raindrops (SV)	NA	3	35
Undeteted vehicles (UV)	51	235	66
False Rate	0.14%	1.01%	1.87%
Average detection range (m)	38.1	37.5	37.4

APPLICATIONS OF LIDAR TRAJECTORIES

All-traffic trajectory data is critical to various traffic research/engineering areas, including but not limited to:

- Connected and autonomous vehicles (CAVs): At present, independent onboard sensing systems do not provide enough information for safe operation in multimodal traffic. An autonomous vehicle with advanced sensors could still be hit by another car on a cross street that fails to stop. To advance safety, vehicles need to obtain the trajectories of all traffic in extended distances, so they can "detect" traffic changes and risks around corners.
- Near-crash analysis: Near-crash events provide essential data for proactive safety analysis and countermeasure recommendation, but this data is difficult to obtain. If all-traffic trajectory data could be collected, we could study vehicle interactions at multiple scales, and define and extract near-crash events to identify traffic safety issues and recommend countermeasures.
- Traffic performance evaluation/adaptive traffic signal control: All-traffic trajectories provide comprehensive information to evaluate traffic performance. Trajectory data reports each road user's stop location, stop time, speed change, and interaction with other road users in addition to conventional vehicle-traffic performance indices such as a number of stops, delay, travel time, and queue length. Optimizing signals along a road is challenging using conventional traffic sensors because system details cannot be accurately observed. Real-time, all-traffic trajectory data can make the traffic system completely observable, thus revolutionizing adaptive traffic control and outperforming conventional systems.
- Automatic pedestrian/wildlife-crossing warning signals: An important application of real-time, all-traffic trajectories is monitoring and predicting vehicle-pedestrian conflicts on urban roads or vehicle-wildlife collision risks on rural highways. Most conventional automatic pedestrian/wildlife warning systems rely on predefined detection areas. These systems trigger warning signals whenever an object is detected in the sensing area, but this has both caused false alarms and failed to identify risks outside the defined areas. Trajectory data tracks the continuous movement of each road user, so crossing detection and prediction can be based on historical trajectory and real-time direction/speed/location for superior accuracy and reliability.

This project studied applications of roadside LiDAR data for pedestrian crossing prediction, detection of animals crossing highways and near-crash identification.

4.1 Detection and prediction of pedestrians crossing roads

In this project, the research team developed a model for pedestrian crossing prediction with the roadside LiDAR sensing system. The developed model provides real-time quantitative confidence-levels of pedestrians crossing roads based on the pedestrian's current and historical trajectories. The prediction employs a modified Naïve Bayes method that was first trained with field LiDAR data (pedestrian movement features and crossing/non-crossing status) and then

evaluated with a testing dataset that is different from the training data. In the training stage, the values of each feature were segmented into ranges, and the combination of the features was used as the input of the Naïve Bayes method.

The Naïve Bayes sequence classifier is a basic generative model (Lewis, 1998) that can be trained efficiently in a supervised learning setting, even with a small dataset (Caruana and Niculescu-Mizil, 2006) so it is suitable for predicting pedestrian crossing behaviors in a real-time way. It requires a strong (naïve) assumption that the features of the sequence data are independent of each other. Velocity and direction features of pedestrian trajectories are discrete and used as the input of the Naïve Bayes classifier. In order to apply the Naïve Bayes algorithm, the value of each feature needs to be segmented into different ranges, and the optimal combination of features was used as the input of the Naïve Bayes model. The training data with labels were divided into several ranges first and checked if the data in each range satisfied the requirement of defined minimum-record. If not, the algorithm reduced the number of segmentations (increases each segmentation's range) and checked the requirement again. The output of the training process is a prediction model that includes the probability information for all the allowable segmentations.

For the prediction process, the inputs are pedestrian trajectories, trained prediction model and probability threshold for predicting pedestrian crossing. The probability threshold is determined by the model user and can be adjusted based on the actual needs. A lower value means higher detection accuracy of pedestrians crossing roads but also a higher probability of misidentifying a non-crossing event as a crossing. A higher threshold value leads to lower detection accuracy of pedestrians crossing roads but also a lower probability of misidentifying a non-crossing event as a crossing. Given a pedestrian's status (a record of the trajectory), the status values can be fitted into one appropriate interval based on the trained model. Since the features are independent of each other, the corresponding probabilities can be multiplied directly. Considering the different segmentations, two maximal probabilities for crossing and non-crossing labels are chosen, and the corresponding segmentations are considered as the optimal segmentations. It cannot be guaranteed that the sum of the two probabilities is equal to one since the chosen segmentations may be different for crossing and non-crossing cases. Therefore, the next step is to normalize two selected maximal probabilities so that the sum of normalized probability values is 1. If the normalized maximal crossing probability is greater than the predefined threshold value, the data point will be assigned a crossing label. Otherwise, the data will be classified into the non-crossing case. In addition, the normalized probabilities are used for providing quantitative confidence level information for crossing and non-crossing predictions.

In this project, 18,000-frame data, streamed from a LiDAR sensor at the intersection of 11th Street and Sierra Street, were processed with the procedures of background filtering, lane identification, road user clustering, pedestrian/vehicle classification, and tracking. Pedestrian trajectory data include XYZ position, total number of points, distance to the LiDAR, tracking ID, frame ID, velocity, direction, and timestamp, and pedestrian/vehicle label information were obtained. Figure 4-1 shows the data collection site on Google Map. The case study only analyzed the pedestrian trajectories on the east side of North Sierra Street. The selected crosswalk is marked with a red rectangle. Three main pedestrian approaching directions were from southbound, northbound, and westbound. Figure 4-2 demonstrates the sample trajectory

data from previous research. The LiDAR sensor is located at the origin (0,0) and not showed in the plot.

Figure 0-20 Intersection for studying pedestrian crossing prediction

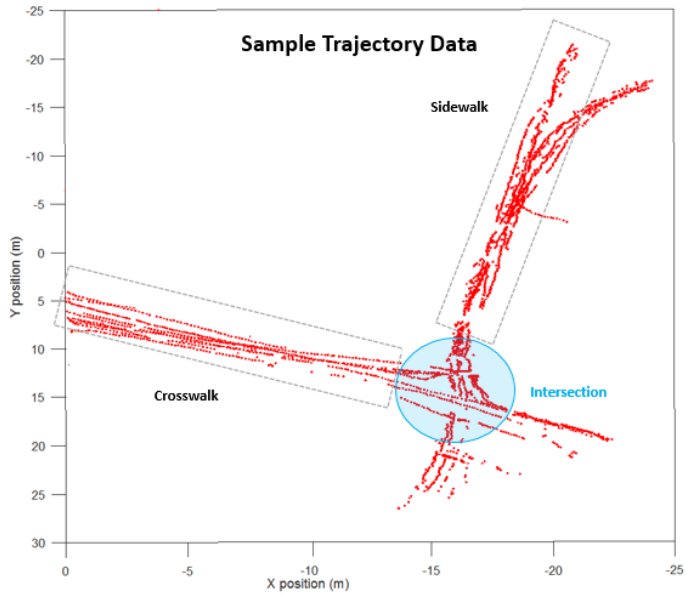


Figure 0-21 Sample trajectories for evaluating pedestrian crossing prediction

The requirement for using Naïve Bayes algorithm is that features in the sequences should be independent of each other. Based on obtained pedestrians' trajectory data, four features – X position (X), Y position (Y), Velocity (V), and Direction (D), were selected for model training. The correlation coefficients between each pair of the above four features are listed in Table 4-1. If the correlation coefficient is less than or equal to 0.2, it means that two variables have no relation or weak relation. The values in the table prove that the selected four features are independent of each other.

Table 0-9 Correlation Coefficients for Selected Features

Features	(X,Y)	(X,V)	(X,D)	(Y,V)	(Y,D)	(V,D)
Correlation Coefficients	0.0101	0.1074	0.0211	0.0125	0.0681	0.1167

According to the trajectory data, 29 non-crossing trajectories and 21 crossing trajectories were used for model training. The trajectories recorded each pedestrian's walking path and status in 8 seconds before actual crossing/non-crossing at the crossing facilities. Four extracted features were X position, Y position, Velocity, and Direction. Based on the proposed modified Naïve Bayes algorithm, a minimal number of segmentation and a maximal number of segmentations were set to one and 10, respectively. The required minimum number of data points in each segmentation was five. Finally, a pedestrian crossing prediction model was trained.

For validating and evaluating the performance of the trained pedestrian crossing prediction model, a testing dataset collected from the same intersection but not seen by the trained model was applied to the trained model. The testing dataset included 10 non-crossing trajectories and 10 crossing trajectories. The probability threshold for crossing warning was 40%. The trained prediction model was directly applied to the testing dataset, and prediction results of a crossing trajectory are demonstrated in Figure 4-3, Figure 4-4, and Table 4-2 as an example. In Figure 4-3, the trajectory shows the pedestrian's walking path from 3 to 0 seconds before the actual crossing. The blue and red dots represent the crossing prediction correctly and incorrectly. Figure 4-4 presents the trend of normalized probability for crossing prediction in terms of time from the start (t). If the total predicted time length is T, then the Time-to-Cross is equal to T - t (e.g., the time before actual crossing). The normalized probability of crossing prediction became increasingly higher as the pedestrian was approaching the crossing facilities. At 3 seconds before crossing, the predicted probability of the pedestrian crossing the road was only 39.9%, while the probability increased to 86.5% at 1 second before crossing. At the moment of actual crossing, the predicted probability for the crossing was 100%, which conforms to common sense.

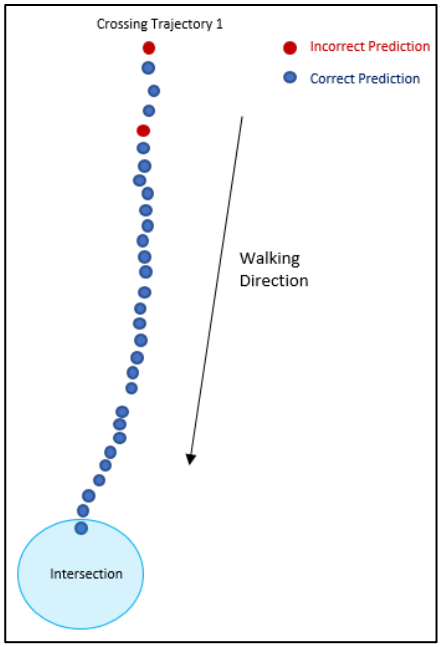


Figure 0-22 Example of crossing-prediction along a pedestrian's trajectory

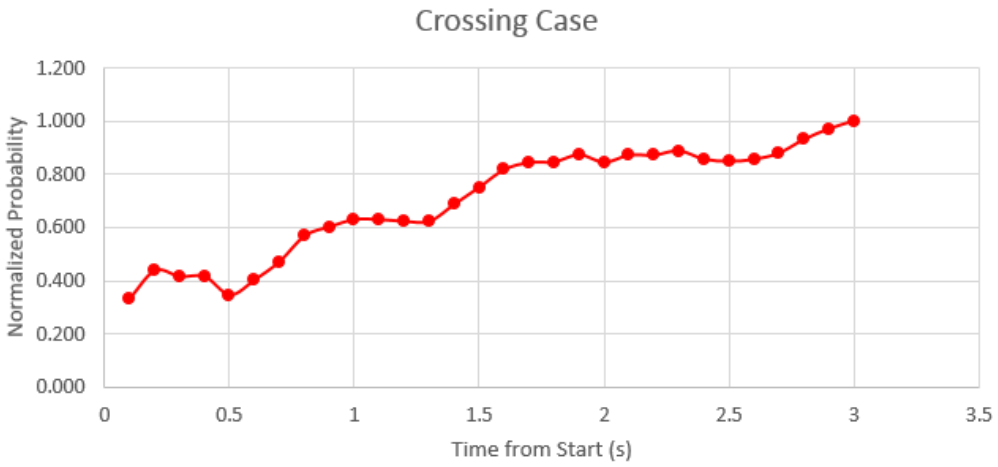


Figure 0-23 Demonstration of predicted-crossing probability changing along a pedestrian's trajectory

Table 0-10 Normalized Probability for Crossing Prediction

Time-to-Cross (second)	3.0	2.5	2.0	1.5	1.0	0.5	0.0
Normalized Probability for Crossing Prediction	39.9%	47.7%	63.7%	82.5%	86.5%	92.6%	100.0%

For all the 10 crossing (label=1) and 10 non-crossing (label=0) trajectories, prediction results are listed in Table 4-3 and Table 4-4. Note that since the crossing warning threshold was set to 40%, there was a higher probability to predict crossing behaviors correctly. The average accuracy for crossing prediction was 97%, and non-crossing prediction accuracy was 84%.

Table 0-11 Prediction accuracy of Ten Crossing Trajectories

Time-to-Cross (second)	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Prediction Accuracy
3.0	0	1	1	1	1	1	1	1	1	1	90%
2.5	1	1	1	1	1	1	1	1	1	1	100%
2.0	1	1	1	1	1	1	1	1	1	1	100%
1.5	1	1	1	1	1	1	1	1	0	1	90%
1.0	1	1	1	1	1	1	1	1	1	1	100%
0.5	1	1	1	1	1	1	1	1	1	1	100%
0.0	1	1	1	1	1	1	1	1	1	1	100%
Total Average Accuracy											97%

Table 0-12 Prediction Accuracy of Ten Non-Crossing Trajectories

Time Before Crossing (second)	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Prediction Accuracy
3	0	0	0	1	0	0	0	0	0	1	80%
2.5	0	0	0	0	0	0	1	0	1	1	70%
2	0	1	0	0	0	0	1	0	0	0	80%
1.5	1	1	0	0	0	0	0	1	0	0	70%
1	0	0	0	0	0	0	0	0	0	0	90%
0.5	0	0	0	0	0	0	0	0	0	0	100%
0	0	0	0	0	0	0	0	0	0	0	100%
Total Average Accuracy											84%

In order to improve prediction accuracy against the influence of random status change among the whole trajectory, three continuous trajectory records (0.3 seconds) were used for aggregated prediction. The final crossing/non-crossing label was determined by combining current prediction and prediction of previous two data-frames. If the goal is to ensure the accuracy of crossing prediction as much as possible, the crossing warning threshold should be decreased. If prediction accuracy for both crossing and non-crossing are sought, the crossing warning threshold should be increased to a relatively high value. The prediction model based on roadside LiDAR trajectories is valuable for unsignalized intersections or mid-block crosswalks. The pedestrian crossing signals, like Rectangular Rapid Flashing Beacon (RRFB), can be triggered automatically based on prediction results and confidence level information. In the future, pedestrian trajectory data in better quality can be obtained by integrating data from multiple LiDAR sensors which are installed at the different locations of the intersection and improved tracking algorithms.

4.2 Detection and Tracking Wildlife Animals Crossing Highway

The testing site for roadside LiDAR detecting and tracking animals crossing a highway is a wildlife overpass (latitude: 40.907406°, Longitude: -114.305021°) on I-80 in Eastern Nevada. The historical data of wildlife animals crossing I-80 through this overpass was used to identify the case study date so that the study could capture wildlife animals crossing the highway in a few days instead of a few weeks. This overpass structure is 200' long and 200' wide over the four I-80 lanes and the median. The exclusionary fence was built on both sides of I-80 to funnel wildlife into the entrance of the overpass. Figure 4-5 shows the wildlife overpass. There is a fence deployed in the middle of the south end of the overpass, which is used to block cows and allow deer to pass.

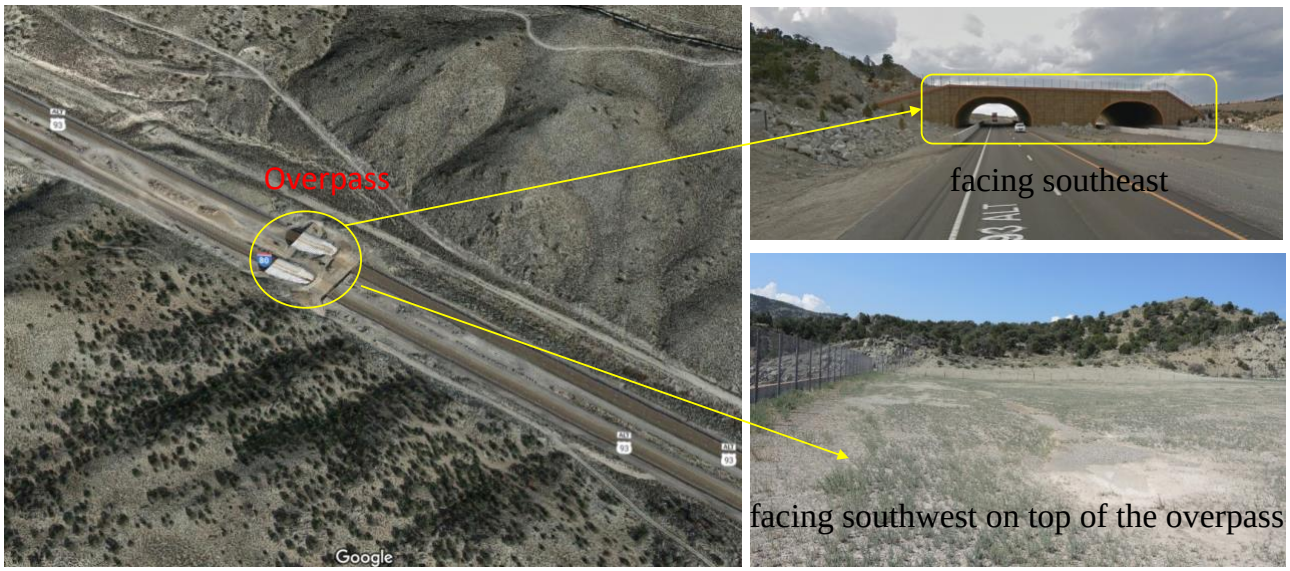
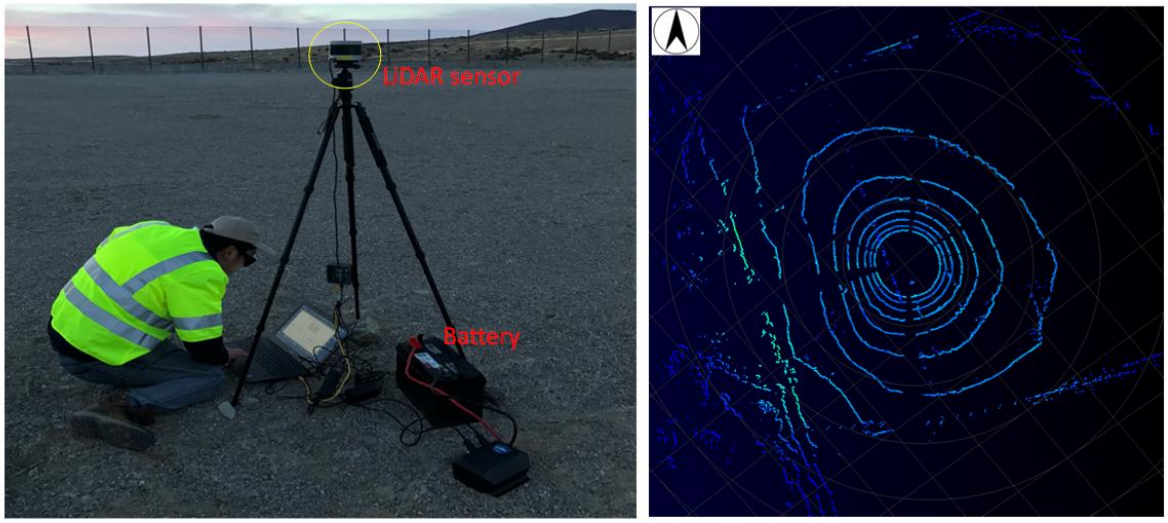


Figure 0-24 Wildlife overpass on 80 in eastern Nevada

The LiDAR sensor was installed at the south end (near the middle of the fence) of the overpass and was mounted on a tripod for temporary data collection and powered by one deep cycle RV battery, as illustrated in Figure 4-6(a). The rotation speed of the LiDAR sensor was set at 10Hz. The data collection time was from 5:00 p.m. on 3/2/2017 to 9:00 a.m. on 3/4/2017 as historical data showed that the frequency of deer using this structure was high in early March every year. A laptop was connected to the LiDAR sensor and used to process the data collected by the sensor. For evaluating the results of LiDAR data processing, a 360-degree camera was also set up near the sensor to help collect the movement of wildlife. Figure 4-7(b) shows an example of a data frame collected by the LiDAR sensor.



(a) Temporary LiDAR installation (b) Example of LiDAR data

Figure 0-25 Animal-crossing data collection site and sample LiDAR data

The collected data showed that eight deer crossed the road using the overpass in the morning of 3/3/2017, from 6:45:36 a.m. to 6:47:46 a.m., as shown in Figure 4-7.

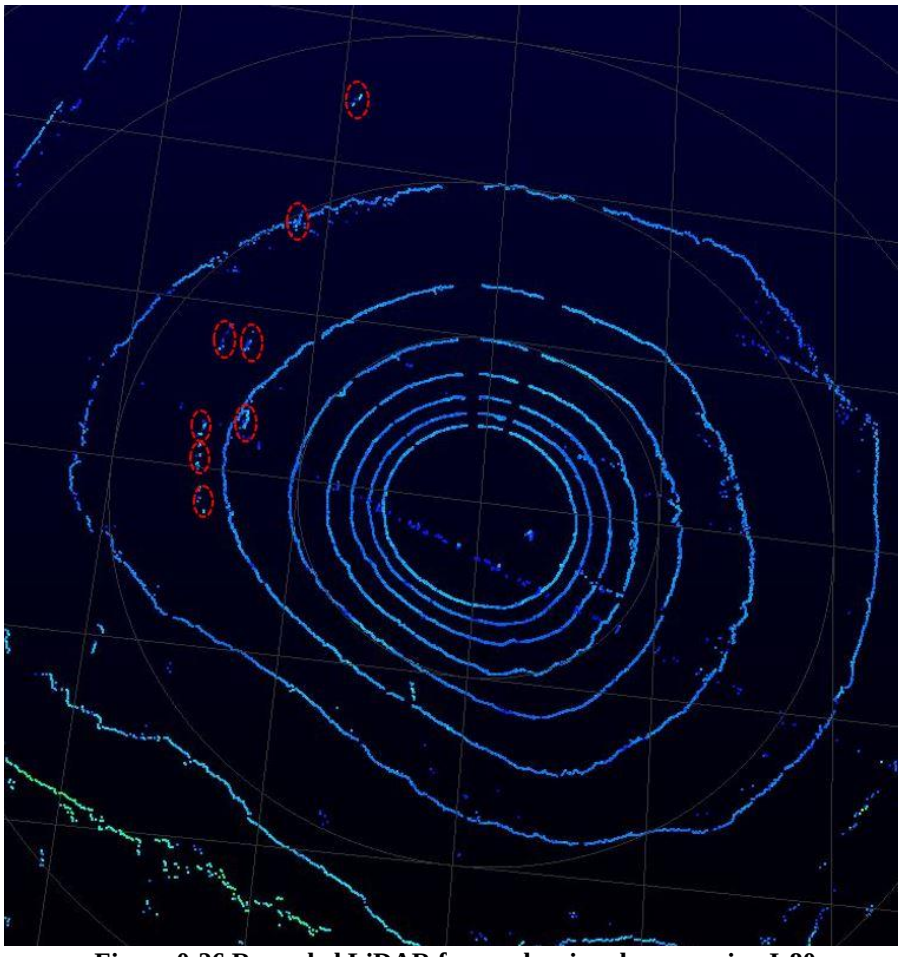


Figure 0-26 Recorded LiDAR frame showing deer crossing I-80

In some frames, only six or seven deer can be detected as some deer may be blocked by the other deer or the middle fence, which may not be successfully captured by the LiDAR sensor. After collecting all the position of wildlife, the trajectories and moving direction of these deer can be generated, as illustrated in Figure 4-8.

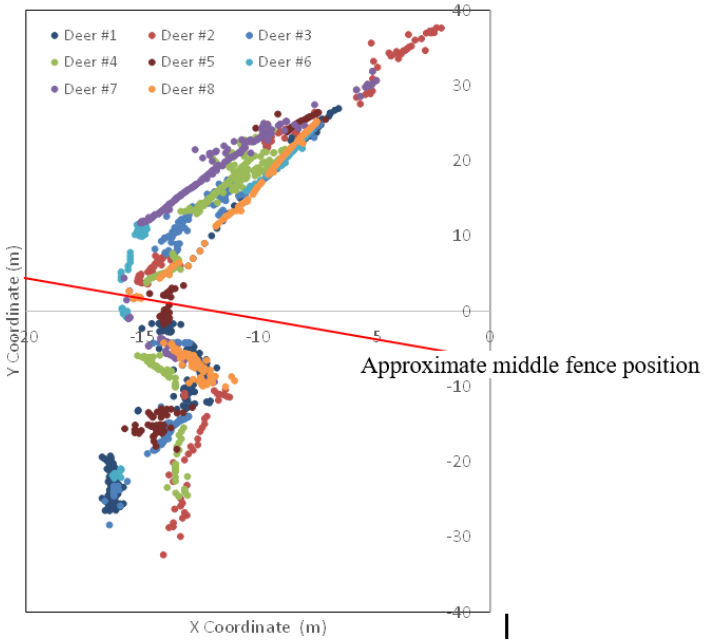


Figure 0-27 Extracted deer trajectories

It can be seen in Figure 4-8, deer changed their direction slightly as they approached the middle fence in order to find a lower section to jump over, which can also be seen in the video uploaded into YouTube: https://www.youtube.com/watch?v=qnvdSN6iusI&feature=youtu.be by the authors. As the deer were gathered near the fence waiting to jump one by one, some deer were blocked by the others or the middle fence. As a result, the beam of the LiDAR sensor could not reach all deer in some frames. The detection range of each deer and average speed calculated are demonstrated in Table 4-5.

Table 0-13 Range of Deer Detection

	Deer #1	**Deer #2**	**Deer #3**	**Deer #4**	**Deer #5**	**Deer #6**	**Deer #7**	**Deer #8**
Max Detection Distance (m)	31.76	37.74	32.83	29.60	27.74	27.51	32.28	26.28
Average Detection Range of All Deer (m)	30.72							
Average Speed of Each Deer (m/s)	5.37	2.77	4.28	4.97	3.65	1.53	4.62	1.97
Average Speed of All Deer (m/s)	3.65							

The average detection range was 30.72 m from the LiDAR sensor. The max effective detection range of the data processing procedure was 37.74 m. Though out of 37.74 m, deer could be seen in the LiDAR data visualization, the developed algorithms could not successfully

identify these points as deer. The average speed of each deer varies from 1.53 m/s to 5.37 m/s, with a total average of 3.65m/s. Since there is a middle fence blocking the overpass, the deer had to slow down to jump the fence and wait for the rest of the herd. Therefore, the total average speed is not high. The wildlife detection in this project provided the method to distinguish wildlife from vehicles; the speed can also be used to distinguish deer and vehicles as the speed limit on the rural area is much higher compared to the speed of deer crossing the road. So, data collection using LiDAR with a longer period is expected to be conducted to capture wildlife crossing the road, which can be used to evaluate the effectiveness of the wildlife and vehicle division method.

4.3 Near-Crash Analysis with Roadside LiDAR Data

Historical crash records are important data sources for safety evaluation on roads. However, a crash is random, and only a small percent of conflicts are actually crashes. Therefore, historical crash data may not accurately reflect the risk of conflicts between road users. Furthermore, delay in safety evaluation is unavoidable since it takes time to collect and process historical crash records. Therefore, researchers and engineers are looking for surrogate data for safety evaluation. Near-crashes are then selected as a surrogate dataset to assess road safety management. Near-crashes refer to cases where drivers execute rapid evasive maneuvers (*i.e.*, emergency braking and/or steering operation) when facing a potential driving risk or a potential threat (Wu and Jovanis, 2013). The major challenge for risk assessment using near-crash data is a data collection method. The data collection method can be roughly divided into three major parts: driving simulation, naturalistic driving studies (NDS) and intelligent transportation systems (ITS). Driving simulation and naturalistic driving data are widely used for near-crash collection in previous studies. Although driving simulators can provide details about near-crashes with low-cost, the driving behavior in simulated environments still differs from a real situation (Wu and Xu, 2017). NDS can collect driver observation and driving operation unobtrusively, which provides a good opportunity to extract near-crashes in the real situation (Wu and Xu, 2018). Near-crashes in NDS were usually identified by detecting unusual vehicle kinematics using accelerometers and gyroscopic sensors installed in the experimental vehicle (Wu and Jovanis, 2013). Although the NDS data have several advantages, those studies still suffered from a major drawback: the NDS data only provide information about the equipped vehicles and their immediate surroundings. For one specific road segment, if there are no vehicles installed with the required NDS devices or the number of vehicles installed with those devices is limited, the near-crashes will be under-reported. Furthermore, the near-crash identification using NDS data is costly since the device installation and data reduction usually require a lot of effort (Xu and Wu 2018). The current strategy for safety analysis can be improved through the fast development of ITS technologies, especially connected-vehicle (CV) technology. In the CV network, the real-time traffic data of all road users can be collected and shared with each other through wireless communication. Currently, there are still limited connected-vehicles on roads, most of which are mainly used for pilot programs. It will take time to equip connected-vehicle devices into all vehicles, especially older models. It was estimated that mixed traffic (i.e. connected vehicles and unconnected vehicles) would exist for the next decade. As a result, not all near-crashes can be detected as only partial vehicle movement information can be obtained. It is necessary to find a solution to fill the data gap during the transition period from unconnected vehicles to connected vehicles. The trajectories of road users provided by roadside LiDAR are considered as the good data source for near-crash

identification.

Different road users may be involved in near-crashes. The major types of near-crashes include vehicle-vehicle near-crash, vehicle-pedestrian near-crash, vehicle-bicycle near-crash, bicycle-pedestrian near-crash, vehicle-animal near-crash, and other types. Previous studies mainly focused on vehicle-vehicle near-crash analysis, especially on rear-end near-crash analysis (Gelso and Sjoberg, 2017). Vehicle-pedestrian near-crash was not well analyzed in previous research, mostly due to the difficulty of vehicle-pedestrian near-crash data collection (Wu and Xu, 2017). Considering the advantages of roadside LiDAR data, this project developed an innovative approach to vehicle-pedestrian near-crash identification using trajectories of vehicles and pedestrians extracted from roadside LiDAR sensors. Detailed thresholds were recommended to define the risk of vehicle-pedestrian conflicts.

A novel method that considers the time difference of reaching the same point between vehicles and pedestrians, the distance between stopped vehicles and pedestrians, as well as the speed-distance profile of vehicles, was developed in this project.

Time Difference to the Point of Intersection

The point of intersection (PI) between trajectories of vehicles and pedestrians can be the potential conflicting point. By comparing the trajectories of vehicles and pedestrians, the location of the PI can be easily extracted. The timestamps when vehicles and pedestrians reach the PI should be different under normal maneuvers. If the timestamps are same, this indicates that a crash happens. Therefore, the time difference to the point of intersection (TDPI) was developed for near-crash identification. TDPI was defined as "the time difference between one vehicle and one pedestrian reaching the same point in their trajectories" in this paper. The TDPI can be calculated through Equation 1:

$$TDPI = ABS\left(\frac{T_v - T_p}{F}\right) \quad (1)$$

Where

TDPI is the time difference between vehicle and pedestrian to the point of intersection (PI), unit: second (s),

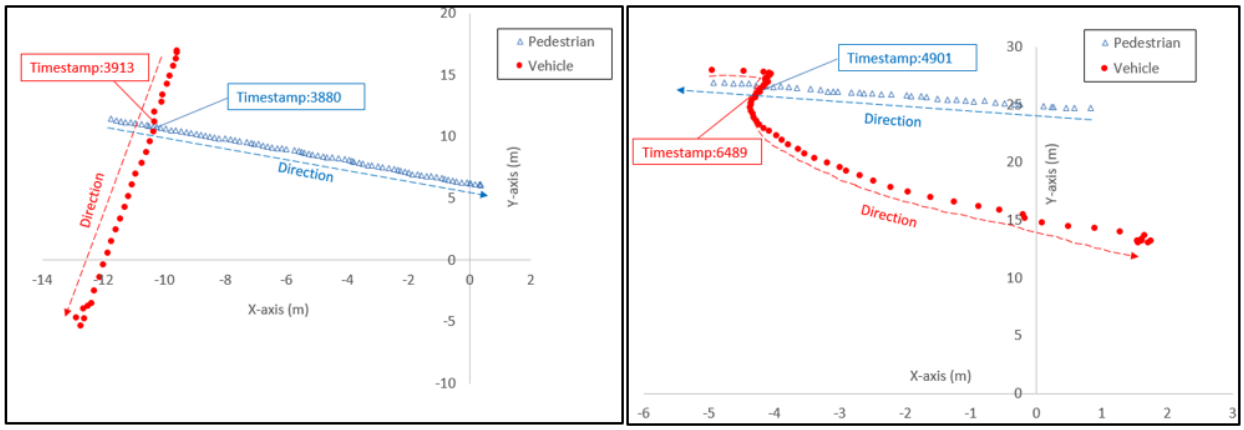
T_v is the timestamp when the vehicle reaches the point of intersection (PI),

T_p is the timestamp when the pedestrian reaches the point of intersection (PI), and

F is the frequency of data collection, unit: HZ.

ABS should also be noted

The TDPI is obtained from the real trajectories without any assumption about speed. A shorter TDPI is noted as more dangerous than a longer one. Figure 4-9 illustrates two examples of different TDPIs.



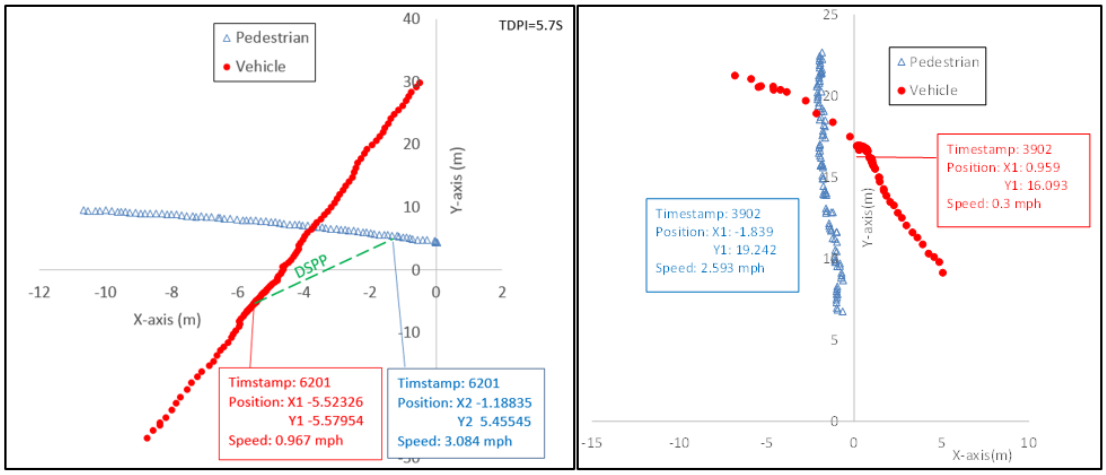
(a) Near-Crash (b) Baseline-Normal Maneuver

Figure 0-28 Examples of TDPI

The frequency of data collection in Figure 4-9 is 10 HZ. In Figure 4-9 (a), one pedestrian reached the PI at the timestamp-3880, and one vehicle reached the PI at the timestamp-3913. In Figure 4-9 (b), one pedestrian reached the PI at the timestamp-3880, and one vehicle reached the PI at the timestamp-4901. The TDPI in Figure 4-9 can be calculated through Equation 1. TDPI in Figure 4-9 (a) is 2.3s, and Figure 4-9 (b) is 148.8s. As can be seen, the situation in Figure 4-9 (a) is more dangerous than that in Figure 4-9 (b) since TDPI in Figure 4-9 (a) is much shorter and closer to driver reaction time. A controlled study in 2000 found the average driver reaction time to brake was 2.3 seconds. A few states, including California, have adopted a standard driver reaction time of 2.5 seconds. Therefore, any event having TDPI less than 2.5 seconds is recommended to be considered as a near-crash incident since the driver reaction time may not be enough to avoid emergency situations. Therefore, the case in Figure 4-9 (a) can be considered as a near-crash. When 2.5s≤TDPI≤3.5s, the crash risk is not as high as those cases when TDPI is less than 2.5 seconds, but pedestrians may feel uncomfortable under the short TDPI. Those cases are considered as crash-relevant, which can still be used for safety assessment when the near-crash events are also limited. If TDPI is higher than 3.5 seconds, the time left for driver reaction is enough, so those cases are considered as normal maneuvers. The thresholds provided in this part are only recommendations based on the authors' best knowledge. Engineers can define their own thresholds for risk assessment.

The distance between Stop Position and Pedestrian

The TDPI may not identify all near-crashes in some cases, such as the driver using the emergency brake and therefore stopping before reaching the PI. Drivers may wait until the pedestrian passes the PI. In that case, the TDPI may be still normal since the conflict occurred before the PI. To address this situation, the distance between the vehicle's stopped position and pedestrian (DSPP) is developed. The DSPP is defined as "the distance between one vehicle and one pedestrian when the vehicle firstly stopped before reaching the pedestrian." Considering the variance in the speed calculation, we select the timestamp when the speed of the vehicle is less than 1.0mph for DSPP calculation. Figure 4-10 shows two examples of different DSPPs with similar TDPIs.



(a) DSPP-11.86m (38.91ft) (b) DSPP-4.21m (13.81ft)

Figure 0-29 Two Different DSPP with Similar TDPI

The DSPP can be then calculated using Equation 2:

$$DSPP = \sqrt{(Xv - Xp)^2 + (Yv - Yp)\text{^}2} \quad (2)$$

Where DSPP is the distance between vehicle and pedestrian when vehicle firstly reduced speed to less than 1.0 mph, unit: m;

Xv is the X-axis of vehicle, unit: meter (s);

Yv is the Y-axis of vehicle, unit: meter (s);

Xp is the X-axis of pedestrian, unit: meter (s);

Yp is the Y-axis of pedestrian, unit: meter (s).

The DSPP in Figure 4-10 (a) is 11.86 m (38.91 ft) and Figure 4-10 (b) is 4.21 m (13.81 ft). The corresponding videos of Figure 4-10 (a) and Figure 4-10 (b) can be reviewed through the following links: https://youtu.be/QObqni4UaSI and https://youtu.be/ovmX6ERaoII, respectively. Though the TDPIs (5.7 s and 5.9 s) in these two cases were similar, the crash risks in those two situations were completely different. Figure 4-10 (a) shows an example that one pedestrian crossed the intersection while drivers stopped far away from the pedestrian. The DSPP was 11.86 m (38.91 ft), which was far enough and safe for the pedestrian. But in Figure 4-10 (b), the left-turn driver did not see the pedestrian crossing the road in advance and stopped in the middle of the intersection when the vehicle was close to the pedestrian. The distance when the drivers stopped at the intersection was only 4.21 m (13.81 ft) away from the pedestrians, which was more dangerous for the pedestrian compared to the case in Figure 4-10 (a). Drivers should stop before the yield line or stop line to give a safe distance to pedestrians. The Manual on Uniform Traffic Control Devices (MUTCD) specified that the distance between the yield/stop line to crosswalk (LTC) should be placed a minimum of 1.2 m (4 ft) in advance of the nearest crosswalk line at the controlled intersection. Stop lines at midblock signalized locations should be placed at least 12 m (40 ft) in advance of the nearest signal indication. If yield or stop lines are used at a crosswalk located at an uncontrolled multi-lane approach, the yield lines or stop lines should be placed 6.1 to 15 m (20 to 50 ft) in advance of the nearest crosswalk line. For normal maneuvers, DSPP should be no shorter than LTC. The recommended thresholds for near-crash identification at different sites are shown in Table 4-6. Engineers can also define their own thresholds based on the features of different sites.

Table 0-14 Near-Crash Identification with DSPP

	Near-Crash	Normal Maneuver
Intersection	DSPP<1.2m (4ft)	DSPP≥1.2m (4ft)
Signalized midblock crosswalk	DSPP<12m (40ft)	DSPP≥12m (40ft)
Uncontrolled midblock crosswalk	DSPP<6.1m (20ft)	DSPP≥6.1m (20ft)

Speed-Distance Profile

The DSPP did not show the impact of different speeds of vehicles on crash risk. For example, two vehicles may have the same DSPPs, indicating they stopped at the same location before the pedestrian. However, they may decrease with different decelerations before the stop. Abrupt stops are noted as more dangerous to pedestrians. The speed-distance profile is used to address this situation. Figure 4-11 shows an example of the speed distribution of vehicles with a different distance of the vehicle to pedestrian (within 100 ft).

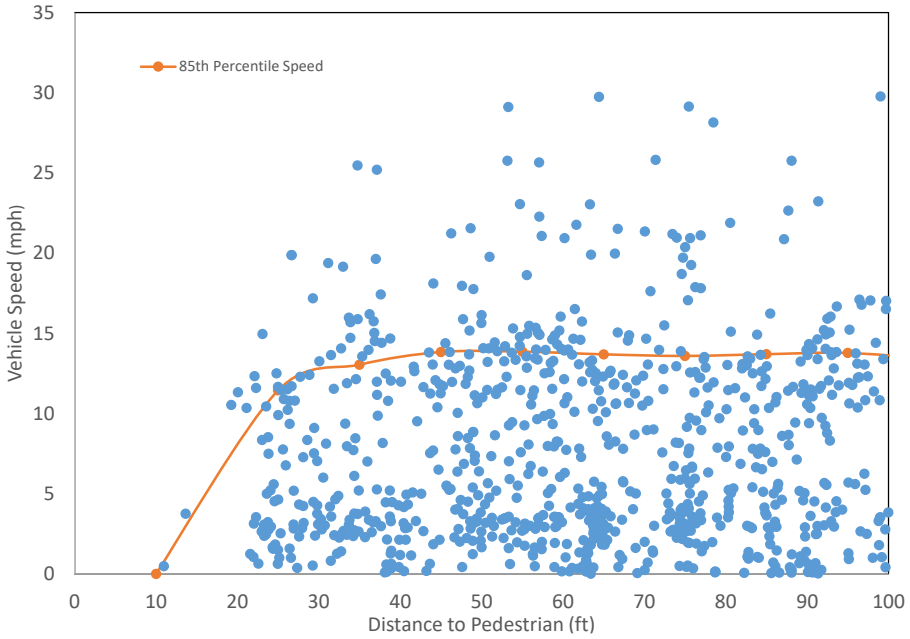


Figure 0-30 Speed-Distance Profile of Vehicles

Most points in Figure 4-11 were located in the area with a distance longer than 20 ft to pedestrians. The distribution of the points was dispersed at the same distance, indicating different vehicles had different speeds. The 85th percentile speed was calculated. In this specific site, the 85th percentile speed was about 15 mph before vehicles dramatically slowed down. The total stopping distance can be estimated by summing the perception-reaction distance and the braking distance. For a vehicle going 15 mph, the estimated stopping distance is 44 ft. The area in Figure 4-11 can be further divided into four subareas using the 85th percentile speed line and the stopping distance, as shown in Figure 4-12. In area A, vehicles had higher speeds within the stopping distance, which were more dangerous to pedestrians compared to those in other areas. Events located in area A can be considered as near-crashes. An example of an event in area A can be found through the link: https://youtu.be/TeEERfWOgzo. In this event, when the distance between the pedestrian and the vehicle was 8 m (26.2 ft), the speed of the vehicle was 20.6

mph. The vehicle did not stop before the pedestrian and passed the pedestrian before the pedestrian finished crossing. For the cases in area B, the vehicles had lower speeds within stopping distance or had higher speeds out of stopping distance, which were considered as crash-relevant. Crash risk in area B was lower than that in area A. An example of an event in area B can be found through the link: https://youtu.be/e9jhlbuk8uw. In this event, the vehicle tried to pass the crossing with high speed when the pedestrian already reached the midblock crosswalk. Though this event may not have the high crash risk like those in area A, the driver's aggressive behavior may leave the pedestrian feeling uncomfortable. In area C, vehicles had lower speeds out of stopping distance, which were considered as safe events to pedestrians. The trajectory of the same vehicle may be located in different subareas in Figure 4-12; the area with the highest crash risk should be used for risk assessment. It should be noted that it is better to check the records in area A manually to make sure they are near-crashes since some points may be the outliers from area B or area C.

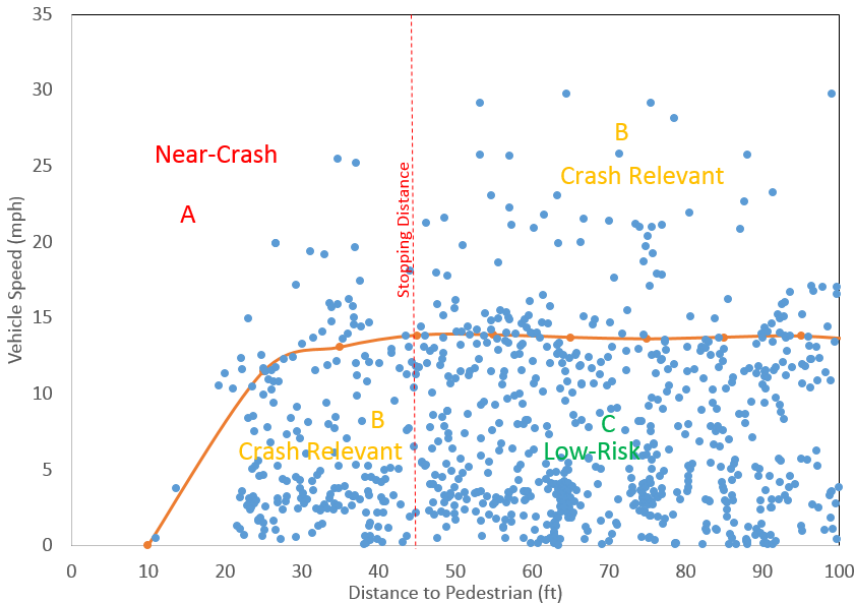


Figure 0-31 An Example of Near-Crash Identification with Speed-Distance Profile

Thresholds of Near-Crash Identification

The TDPI, DSPP and Speed-Distance profile can all be used for near-crash identification. The final thresholds were determined by combining these three factors. All events can be divided into three parts based on their risks: near-crash, crash-relevant, and low risk. Table 4-7 shows the final recommended thresholds for near-crash identification. This algorithm has been coded into an automatic procedure in Matlab for near-crash identification.

Table 0-15 Near-Crash Identification

Risk	Thresholds
Near-crash	TDPI<2.5s or 0<DSPP<LTC or vehicle speed within area A in speed-distance profile
Crash Relevant	2.5s≤TDPI≤3.5s or vehicle speed within area B in speed-distance profile
Low risk	TDPI>3.5s or DSPP≥LTC or vehicle speed within the area (a) in speed-distance profile within area C in speed-distance profile

For evaluating the applicability of the new algorithm, two case studies were conducted. The following criteria are recommended to select the strategic sites for installing LiDAR: 1) The sites with high historical pedestrian-involved crash frequency. 2) The sites where the public has concern or complainst about pedestrian safety. 3) New opened intersections (need design and operation evaluation). In this project, the roadside LiDAR data were collected at two sites in Reno, Nevada.

One site was selected at one midblock on 15th St. by the campus of the University of Nevada, Reno (UNR). There is a steep slope at this site. Many students and faculties at the university have a safety concern about this midblock. The other site, N. Virginia St. and10th St., is an intersection with Rectangular Rapid Flash Beacon (RRFB). A vehicle-pedestrian crash happened at this intersection on July 19th, 2017, which was reported in the following link: http://mynews4.com/news/local/reno-police-investigate-vehicle-pedestrian-crash-near-unr-campus. Further examination was required to evaluate pedestrian safety at this site. The peak hour data were collected on the same weekday. The collection time was 25 minutes (15000 frames) at both sites. Figure 4-13 and Figure 4-14 show the results of TDPI, DSPP and speed-distance profile at the two sites.

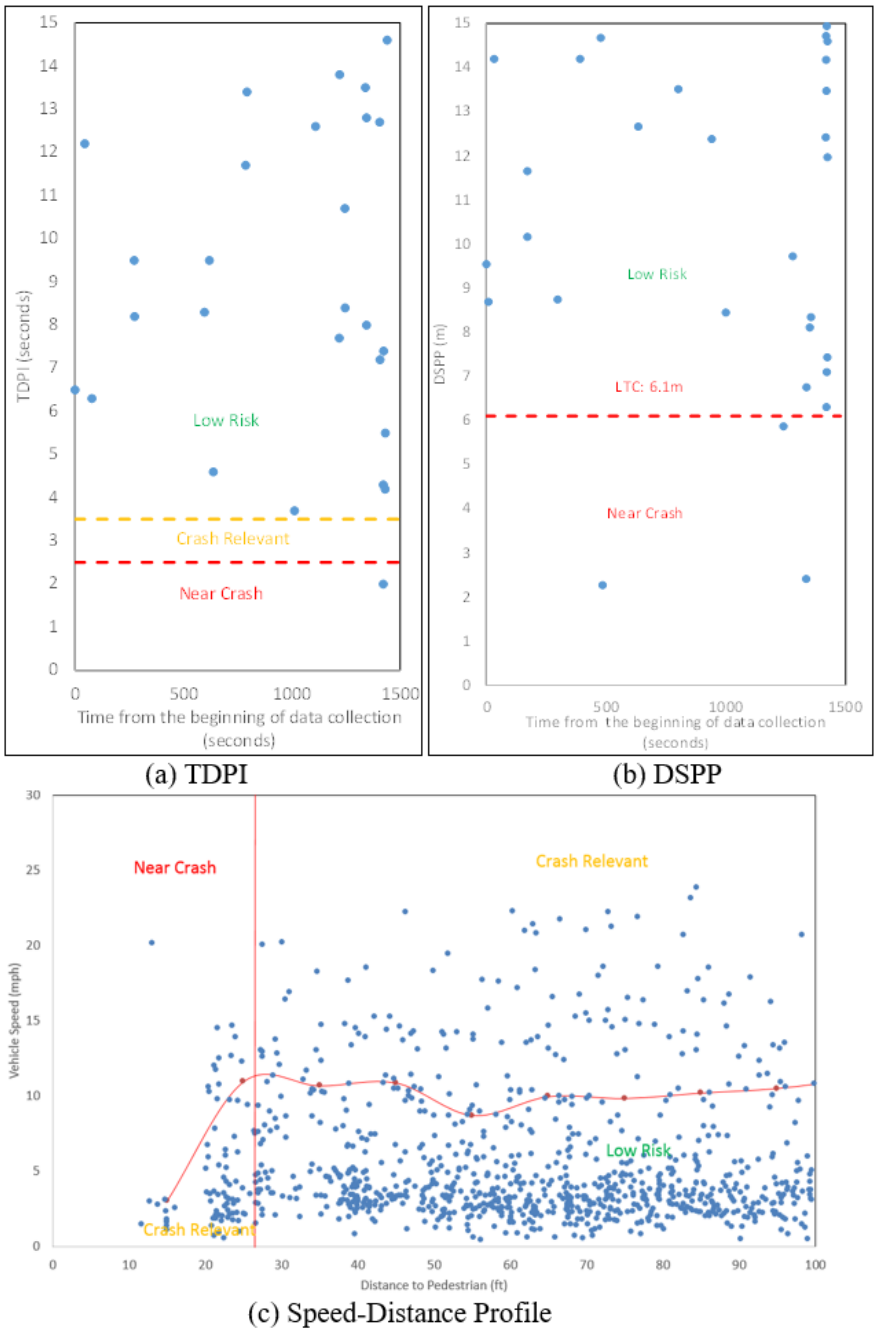


Figure 0-32 Near-Crash identification at One Midblock on 15th Street

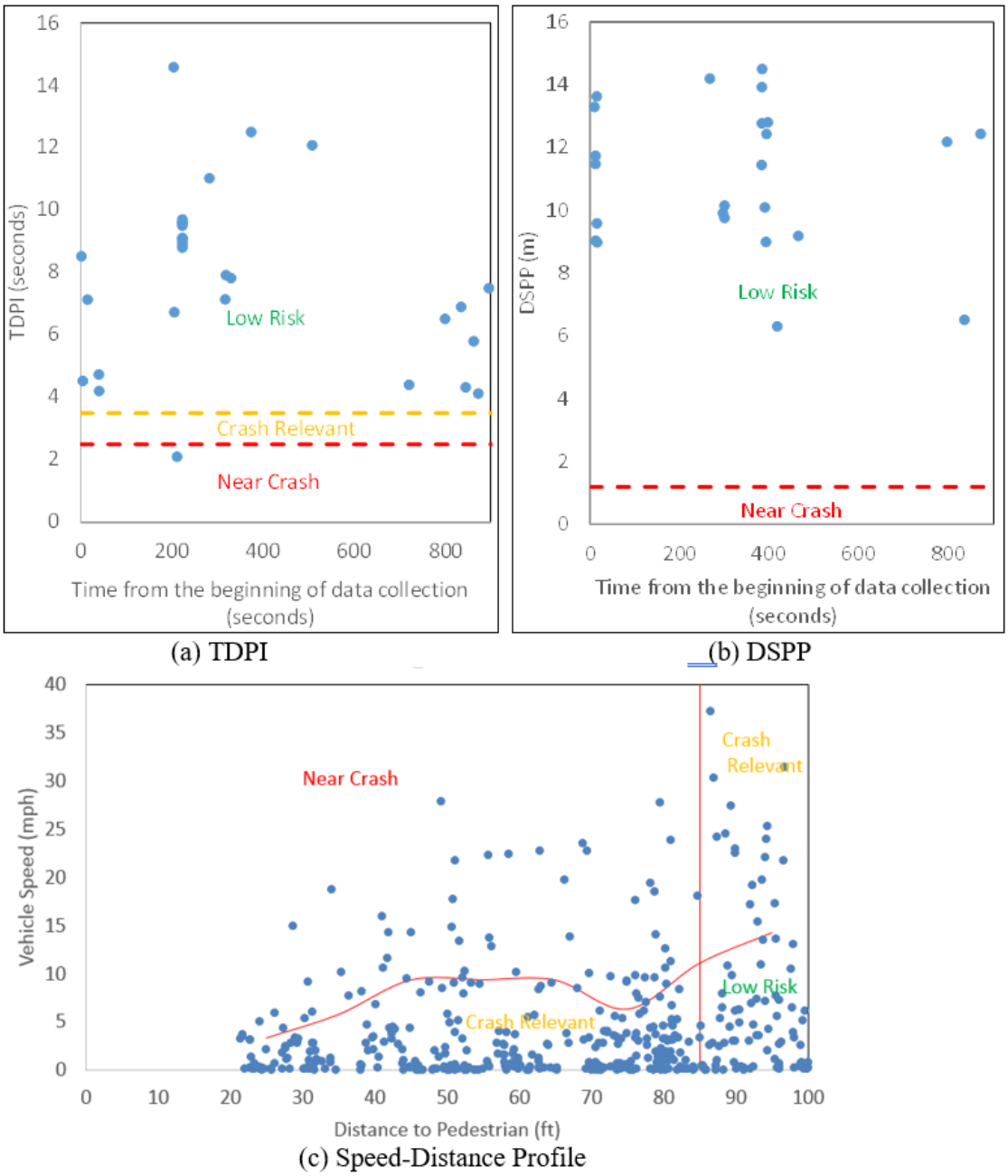


Figure 0-33 Near-Crash identification at N. Virginia St. and 10th St.

An example of a near-crash (the event with a 2.1-seconds of TDPI in Figure 4-13 (a)) incident that occurred at the second site was shown in the following video: https://youtu.be/-3-tgqeQsHE. The video showed that the driver failed to yield to the pedestrian. By checking the videos, the near-crash events can be confirmed. The results of case studies show that the vehicle-pedestrian near-crash events can be identified using roadside LiDAR trajectory data.

CONCLUSIONS AND RECOMMENDATIONS

This project developed methodologies for obtaining high-accuracy all-traffic trajectories with 360-degree LiDAR sensors deployed at the roadside. The data processing procedure provides trajectory-level movement status of all road users in the sensing range that can be an intersection, an arterial or a city road network with roadside LiDAR sensors deployed. The all-traffic trajectory data are essential to connected and autonomous vehicles to perceive traffic situations or threats outside the "sight of view" of onboard sensing systems. The data could also bring revolutions to conventional traffic engineering areas. For example, accurate pedestrian trajectories can be used for pedestrian behavior studies to improve pedestrian safety; Roadside LiDAR can trigger pedestrian signals automatically according to real-time location, speed, and direction information. Conclusions and recommendations related to the project findings were summarized as follows.

Background filtering

The 3D-DSF method developed in this study filters both static background and vibrating background efficiently. This algorithm learns the threshold of cube-point-density (TD) automatically with only two required inputs: the number of aggregated frames and the side length of cubes. Recommended values for these two parameters were suggested in this project based on sensitivity analysis. Case study results proved that the 3D-DSF method well served background-filtering under different road situations (high traffic volume; night time; different speed limits). After obtaining the 3D matrix that labels background areas, real-time background-filtering can exclude uninterested data points so that the accuracy and efficiency of the following steps can be significantly improved.

Though there were still noise laser points left after applying 3D-DSF, the clustering results showed that the scattered noise points did not reduce the accuracy of object identification. In future research, it is expected to test other machine learning or artificial intelligence methods to improve the accuracy further. For redressing the background, it is recommended to run the algorithm in the backstage every several hours to update any changes of environment, which can also improve background-filtering results.

Clustering, classification and tracking

The case studies of LiDAR data processing showed that detection accuracy of object clustering was about 96%. The main reason for clustering failure was that low-density points from objects at a far distance. Occlusion was another issue. An approximate 96% classification accuracy can be achieved within the 30-meter-radius sensing range. The classification failure was also found to be caused by vehicle occlusion. Tracking accuracy of approximately 95% can be realized within about 30 m detection radius from the LiDAR sensor. The effective distance was done by VLP-16 LiDAR sensors, while the project team achieved doubled effective-distance with VLP-32 LiDAR sensors. The statistic results showed that 90% of the speed values were with an offset lower than 2.5mph.

For future work, methods of object detection and tracking can be extended to bicycles in

urban areas and wildlife in rural areas. Algorithms will be further validated and improved for accuracy and reliability of different traffic scenarios. Multiple LiDAR sensors will be deployed at intersections and along roads to cover a larger area and reduce the possibility of occlusion. The extracted data will be applied and tested in connected-vehicle applications and other practices for traffic safety, mobility, and efficiency.

Pedestrian-crossing prediction

This project developed a modified Naïve Bayes crossing prediction model to predict pedestrian crossing roads (at a crosswalk or not) by using the trajectory data from roadside LiDAR. This model provides real-time quantitative confidence level information for predicting pedestrian crossing behaviors using trajectory-level movement features. The major steps include pedestrian tracking, pedestrian feature extraction, prediction model training, and evaluation. A comprehensive case study showed the effectiveness of the proposed algorithm using the real data, and the trained modified Naïve Bayes prediction model has a higher accuracy in crossing prediction than the basic ANN model. Besides, the confidence level information with adjustable key parameters makes it possible to use the proposed prediction model for real-world applications. The proposed prediction method provides an innovative way to understand pedestrians' crossing behaviors. The prediction model is valuable for detection of unsignalized intersections or mid-block crossing. The pedestrian crossing signals like Rectangular Rapid Flashing Beacon (RRFB), can be automatically triggered by roadside LiDAR after the trajectories are processed and predicted. It can also advance connected and autonomous vehicle fields by providing real-time warning/alert messages to vehicles and pedestrians, thus reducing the probabilities of vehicle-to-pedestrian crashes.

In the future, pedestrian trajectory data in better quality can be obtained by integrating data from multiple LiDAR sensors and improved algorithms. Advanced prediction models with higher accuracy and lower computational expense still need to be explored. Warning of wildlife crossings may also be an application of crossing predictions.

Detection of animal crossing

The field test shows that the methods (using VLP-16) could detect wildlife with a max radius of 37.74 m (124 ft) around the LiDAR sensor. The detection range varies from deer subject to different distances and angles from the LiDAR. The detection range can be extended by using sensors with more laser beams or deploying and integrating multiple LiDAR sensors in the detection area. The developed technology can provide real-time position, moving direction and speed information of animals, which can be used to trigger flashing warning signs to warn approaching drivers. This innovative sensing approach can also be used to analyze wildlife behavior, such as what is the peak time used by wildlife to cross the road and which route was preferred. The system can also be deployed along different crossing structures to evaluate the effect of crossing structures and understand wildlife-crossing patterns. The same sensors and methods can also be used to detect and track rural vehicle traffic. The wildlife information and vehicle trajectories can be integrated and broadcasted through the connected vehicle communication system to support future autonomous and connected vehicles.

Application of roadside LiDAR for detecting and tracking wildlife animals crossing

highways can be improved by enhancing current data processing algorithms. In this pilot study, the sensor was powered by batteries that need to be charged every day. In the future, it is recommended to power the roadside sensor and computer with solar power devices and batteries. As mentioned before, this pilot application detected deer that were crossing I-80 through a wildlife overpass instead of crossing the road surface, so a data collection with a more extended period is expected to be conducted to capture animals walking on the road surface.

Near-crash analysis

This project developed a novel method for vehicle-pedestrian near-crash identification using the trajectories of vehicles and pedestrians extracted from roadside LiDAR data. Three factors: TDPI, DSSP, and speed-distance profile were combined for vehicle-pedestrian near-crash identification. The proposed method was coded into an automatic procedure to release the heavy labor work for near-crash identification. The case studies showed that the crash risk could be measured by analyzing trajectories from LiDAR without waiting for historical crash records. Though this paper provided recommended parameters for vehicle-pedestrian near-crash identification, method users can select their thresholds based on different features of specific sites. The proposed method could be a critical safety performance measure for before-and-after pedestrian safety assessment or can be used to identify a site with the highest pedestrian-crash risk from the site pool.

Improvement of near-crash event analysis is expected in further studies. Bicycle-involved near-crash was not considered in this paper since the bicycle data were limited. Videos were not recorded in the two case studies. In the next step, videos will be used to validate the near-crashes. The two case studies provided preliminary validation of identifying near-crashes using roadside LiDAR data. More data are expected to be collected to ensure the systematic validation of the near-crashes in further studies. The implementation of the proposed method relies on the accurate trajectories of vehicles and pedestrians. The previous sections mentioned that object occlusion was a primary reason for this error. Setting up multiple LiDARs in different directions are expected to solve this issue.

REFERENCES

Aggarwal, C.C. and Reddy, C.K. eds., 2013. Data clustering: algorithms and applications. CRC press.

Allodi, M., Broggi, A., Giaquinto, D., Patander, M. and Prioletti, A., 2016, June. Machine learning in tracking associations with stereo vision and lidar observations for an autonomous vehicle. In Intelligent Vehicles Symposium (IV), 2016 IEEE(pp. 648-653). IEEE.

Azim, A. and Aycard, O., 2012, June. Detection, classification and tracking of moving objects in a 3D environment. In Intelligent Vehicles Symposium (IV), 2012 IEEE (pp. 802-807). IEEE.

Azim, A. and Aycard, O., 2012, June. Detection, classification and tracking of moving objects in a 3D environment. In Intelligent Vehicles Symposium (IV), 2012 IEEE (pp. 802-807). IEEE.

Belton, D., Moncrieff, S. and Chapman, J., 2013. Processing tree point clouds using Gaussian mixture models. Proceedings of the ISPRS annals of the photogrammetry, remote sensing and spatial information sciences, Antalya, Turkey, pp.11-13.

Brackstone, M., McDonald, M. and Sultan, B., 1999. Dynamic behavioral data collection using an instrumented vehicle. Transportation Research Record: Journal of the Transportation Research Board, (1689), pp.9-16.

Caruana, R. and Niculescu-Mizil, A., 2006, June. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168). ACM.

Chehata, N., David, N. and Bretar, F., 2008, July. LIDAR data classification using hierarchical K-means clustering. In ISPRS Congress Beijing 2008 (Vol. 37, No. B3b, pp. 325-330).

Cheng, J., Xiang, Z., Cao, T. and Liu, J., 2014, May. Robust vehicle detection using 3D Lidar under complex urban environment. In Robotics and Automation (ICRA), 2014 IEEE International Conference on (pp. 691-696). IEEE.

Coifman, B., Beymer, D., McLauchlan, P. and Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies, 6(4), pp.271-288.

Csanyi, N. and Toth, C.K., 2007. Improvement of lidar data accuracy using lidar-specific ground targets. Photogrammetric Engineering & Remote Sensing, 73(4), pp.385-396.

Elmqvist, M., 2001, March. Ground estimation of laser radar data using active shape models. In OEEPE workshop on airborne laserscanning and interferometric SAR for detailed digital elevation models (pp. 1-3).

Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

Gelso, E.R. and Sjoberg, J., 2017. Consistent threat assessment in rear-end near-crashes using BTN and TTB metrics, road information and naturalistic traffic data. IEEE Intelligent Transportation Systems Magazine, 9(1), pp.74-89.

Gupta, S., Weinacker, H. and Koch, B., 2010. Comparative analysis of clustering-based approaches for 3-D single tree detection using airborne fullwave lidar data. Remote Sensing, 2(4), pp.968-989.

Himmelsbach, M., Mueller, A., Lüttel, T. and Wünsche, H.J., 2008, October. LIDAR-based 3D object perception. In Proceedings of 1st international workshop on cognition for

technical systems (Vol. 1).

Ismail, K., Sayed, T., Saunier, N. and Lim, C., 2009. Automated analysis of pedestrian–vehicle conflicts using video data. Transportation research record, 2140(1), pp.44-54.

Kidono, K., Miyasaka, T., Watanabe, A., Naito, T. and Miura, J., 2011, June. Pedestrian recognition using high-definition LIDAR. In Intelligent Vehicles Symposium (IV), 2011 IEEE (pp. 405-410). IEEE.

Lee, H. and Coifman, B., 2012. Side-fire lidar-based vehicle classification. Transportation Research Record, 2308(1), pp.173-183.

Lewis, D.D., 1998, April. Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning (pp. 4-15). Springer, Berlin, Heidelberg.

Li, Q., Zheng, N. and Cheng, H., 2004. Springrobot: A prototype autonomous vehicle and its algorithms for lane detection. IEEE Transactions on Intelligent Transportation Systems, 5(4), pp.300-308.

LiDAR, V., 2016. VLP-16 In VLP-16 Manual: User's Manual and Programming Guide; Velodyne LiDAR. Inc.: San Jose, CA, USA.

Miyasaka, T., Ohama, Y. and Ninomiya, Y., 2009, June. Ego-motion estimation and moving object tracking using multi-layer lidar. In Intelligent Vehicles Symposium, 2009 IEEE (pp. 151-156). IEEE.

Morsdorf, F., Meier, E., Kötz, B., Itten, K.I., Dobbertin, M. and Allgöwer, B., 2004. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. Remote Sensing of Environment, 92(3), pp.353-362.

Morton, P., Douillard, B. and Underwood, J., 2011, December. An evaluation of dynamic object tracking with 3D LIDAR. In Proc. of the Australasian Conference on Robotics & Automation (ACRA).

Mukhtar, A., Xia, L. and Tang, T.B., 2015. Vehicle Detection Techniques for Collision Avoidance Systems: A Review. IEEE Trans. Intelligent Transportation Systems, 16(5), pp.2318-2338.

Premebida, C., Ludwig, O. and Nunes, U., 2009, October. Exploiting lidar-based features on pedestrian detection in urban scenarios. In Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on (pp. 1-6). IEEE.

Premebida, C., Monteiro, G., Nunes, U. and Peixoto, P., 2007. A lidar and vision-based approach for pedestrian and vehicle detection and tracking. rn, 10, p.2.

Reina, G., Underwood, J., Brooker, G. and Durrant-Whyte, H., 2011. Radar-based perception for autonomous outdoor vehicles. Journal of Field Robotics, 28(6), pp.894-913.

Reynolds, D., 2015. Gaussian mixture models. Encyclopedia of biometrics, pp.827-832.

Sivaraman, S. and Trivedi, M.M., 2013. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Transactions on Intelligent Transportation Systems, 14(4), pp.1773-1795.

Spinello, L. and Siegwart, R., 2008, May. Human detection using multimodal and multidimensional features. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on (pp. 3264-3269). IEEE.

Szarvas, M., Sakai, U. and Ogata, J., 2006, June. Real-time pedestrian detection using LIDAR and convolutional neural networks. In Intelligent Vehicles Symposium, 2006 IEEE (pp. 213-218). IEEE.

Vosselman, G., 2000. Slope based filtering of laser altimetry data. International Archives of Photogrammetry and Remote Sensing, 33(B3/2; PART 3), pp.935-942.

Wang, D.Z., Posner, I. and Newman, P., 2012, May. What could move? finding cars, pedestrians and bicyclists in 3d laser data. In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 4038-4044). IEEE.

Wang, H., Wang, B., Liu, B., Meng, X. and Yang, G., 2017. Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. Robotics and Autonomous Systems, 88, pp.71-78.

Wojke, N. and Häselich, M., 2012, May. Moving vehicle detection and tracking in unstructured environments. In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 3082-3087). IEEE.

Wu, J. and Xu, H., 2017. Driver behavior analysis for right-turn drivers at signalized intersections using SHRP 2 naturalistic driving study data. Journal of safety research, 63, pp.177-185.

Wu, J. and Xu, H., 2018. The influence of road familiarity on distracted driving activities and driving operation using naturalistic driving study data. Transportation research part F: traffic psychology and behaviour, 52, pp.75-85.

Wu, J., Xu, H., Sun, Y., Zheng, J. and Yue, R., 2018. Automatic Background Filtering Method for Roadside LiDAR Data. Transportation Research Record, p.0361198118775841.

Wu, K.F. and Jovanis, P.P., 2013. Screening naturalistic driving study data for safety-critical events. Transportation research record, 2386(1), pp.137-146.

Xu, H. and Wu, J., 2018. Use of Naturalistic Driving Study Data to Determine Right-Turn Driver Deceleration Behavior at Signalized Intersections (No. 18-01877).

Yang, B., Dong, Z., Zhao, G. and Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 99, pp.45-57.

Yang, B., Fang, L. and Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 79, pp.80-93.

Yang, B., Huang, R., Dong, Z., Zang, Y. and Li, J., 2016. Two-step adaptive extraction method for ground points and breaklines from lidar point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 119, pp.373-389.

Zeng, X., Tao, C. and Chen, Z., 2009, December. The application of DSRC technology in intelligent transportation system. In Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference on (pp. 265-268). IET.

Zheng, J., Wang, Y., Nihan, N. and Hallenbeck, M., 2006. Extracting roadway background image: Mode-based approach. Transportation Research Record: Journal of the Transportation Research Board, (1944), pp.82-88.