3D urban traffic monitoring via roadside multi-beam lidar

Jiaxing Zhang

Thesis submitted for the degree of Doctor of Philosophy

Newcastle University

School of Engineering

Abstract

Smart mobility, one of the most important topics in smart city studies, aims to reduce pollution, mitigate traffic congestion and enhance safety through research into Mass Transit Systems, Individual Mobility, and Intelligent Transportation Systems (ITS). ITS are advanced applications to collect, store and process data, information and knowledge aimed at planning, implementing and evaluating integrated initiatives and policies of smart mobility. Within ITS, high-resolution microscopic traffic data (HRMTD) can be obtained by infrastructure-based monitoring systems relying on various types of sensors. In the context of traffic monitoring, the acquisition of comprehensive information presents challenges in vehicle detection, tracking, reconstruction and classification. However, many existing traffic monitoring studies cover only one or two of these challenges, and the related developments are either not state-of-the-art or inapplicable to the relatively new technology of lidar (light detection and ranging) systems that are capable of acquiring accurate 3D data in real-time for future urban traffic monitoring.

This research develops a 3D lidar-based traffic monitoring system that can provide comprehensive information through an end-to-end workflow, thereby determining fundamental traffic parameters including the number of vehicles, vehicle dynamics, dimensions and types. A three-step method is employed to realize vehicle detection, in which the first two steps are moving points extraction and instance clustering. The final step, vehicle and non-vehicle classification, is implemented by both a deep learning method (PointVoxel-RCNN, PV-RCNN) and a traditional machine learning approach (Random Forest, RF). Two frameworks are proposed to perform vehicle tracking. The first aims to provide more accurate vehicle speeds via a tracking refinement module. The other runs tracking and detection in parallel so that misdetections from the vehicle detection stage can be mitigated. Vehicle reconstruction is then implemented from the perspectives of both 2D and 3D without assuming any a priori knowledge. Vehicles can be fine-grained classified into different categories such as car, van, bus and truck.

The developed traffic monitoring system has been practically demonstrated using data acquired from different laser scanners operating in different urban scenarios. It has been evaluated using roadside lidar data obtained from two different panoramic 3D lidar sensors, a RoboSense RS-

LiDAR-32 and a Velodyne VLP-16, in four real-world case studies: a road section including a round corner, a straight road section near a traffic light, a road junction and a crossroad, respectively. Based on experimentation, more than 94 % of on-road vehicles are detected and tracked with a mean speed accuracy of 0.2m/s. The average range of vehicle trajectories is increased by c. 21% from the results of different scenes based on the improved framework. The continuity of the trajectories is also enhanced and the maximum effective tracking ranges of both tested laser scanners in different traffic scenes are found to exceed 110m. The dimensions of the vehicles being reconstructed are assessed with a Root Mean Square Error (RMSE) smaller than 0.24m. Vehicles are further classified into different categories with F_1 score greater than 0.90. The reported accuracies demonstrate the potential of the developed system to efficiently serve fine-grained urban traffic monitoring.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Jon Mills and Dr. Wen Xiao, even though there will never be enough words, for their support, encouragement, advice and assistance throughout the hard years of my PhD. I am very grateful to them for having faith in me. This thesis would not have been possible without their help and support. Many special thanks to their deep concerns for me during the pandemic.

I would like to thank the China Scholarship Council (CSC) and Newcastle University for funding my PhD program over the past four years. I would also thank UKCRIC - UK Collaboratorium for Research in Infrastructure & Cities: Newcastle Laboratories for supporting this PhD project.

My appreciation due to the Department of Archaeology for loan of the Routescene instruments used in this research. My sincere thanks also go to Prof. Benjamin Coifman from Ohio State University for his guidance and support in data collection and experiment design in the first year. His insightful thoughts about lidar have indeed broadened my horizon. I am deeply grateful to Mr. Martin Robertson, Mr. James Goodyear and Mr. Nathan Harrap for their timely technical support. I would also thank Daniel Bell, Maria Peppa and Ahmed Elsherif in the Geospatial Engineering group for all their help and guidance in data collection and processing. I also deeply appreciate the guidance in deep learning theory and help in workstation utilization from Dr. Shidong Wang. Many thanks to Sven Berendsen for helping me with some tough IT issues.

I would like to thank the entire Geospatial Engineering group at Newcastle University for making me feel at home in Newcastle. Special thanks to my friends Chen Yu, Chuang Song, Zheng Wang, Hui Luo, Lesley Davidson, Marine Roger, Gauhar Meldebekova, Katarina Vardic, etc., for their help and all the good memories we had. I am also eager to extend my gratitude to all the friends I met in Newcastle.

Lastly, I would like to thank my parents, my brother and my sister-in-law for their unconditional love, support and understanding throughout my study, which is always the power to help me move forward. I would also thank my friends in China for their help and encouragement when I am far away from home.

Related Publications

The following publications contain work related to or derived from this thesis:

Zhang, J., Xiao, W., Coifman, B. and Mills, J., 2019. Image-based vehicle tracking from roadside lidar data. *ISPRS Geospatial Week* 2019.

Zhang, J., Xiao, W., Coifman, B. and Mills, J.P., 2020. Vehicle tracking and speed estimation from roadside lidar. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, pp.5597-5608.

Zhang, J., Xiao, W. and Mills, J.P., 2022. Optimizing Moving Object Trajectories from Roadside Lidar Data by Joint Detection and Tracking. *Remote Sensing*, 14(9), p.2124.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3D-DSF	3D Density-Statistic-Filtering
3DSSD	3D Single Stage object Detector
ANPR	Automatic Number Plate Recognition
AP	Average Precision
BP-ANN	Back Propagation Artificial Neural Network
BP-NN	Back Propagation Neural Network
CCTV	Closed-Circuit Television
CDBNs	Convolutional Deep Belief Networks
CDF-HC	cumulative distribution functions Havrda-Charvát
CNN	Convolutional Neural Network
CV	Connected Vehicle
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DSLR	Digital Single-Lens Reflex
D-FPS	Furthest-Point-Sampling based on 3D Euclidean distance
ECE	Euclidean Cluster Extraction
EKF	Extended Kalman Filter
EM	Expectation-Maximisation
FOV	Field-Of-View
FP	False Positive
FPS	Furthest Point Sampling
F-FPS	Furthest-Point-Sampling based on Feature distance
GMM	Gaussian mixture models
GMPHD	Gaussian Mixture Probability Hypothesis Density
GNN	Global Nearest Neighbor
GNSS	Global Navigation Satellite System
GT	Ground Truth
HRMTD	High Resolution Microscopic Traffic Data
ICCP	Iterative Closest Compatible Point

ICP	Iterative Closest Point
ICPIF	Iterative Closest Point using Invariant Features
ICT	Information and Communication Technology
IoT	Internet of things
IOU	Intersection Over Union
IMU	Inertial Measurement Unit
ITS	Intelligent Transportation Systems
JDAT	Joint Detection And Tracking
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
JRMPC	Joint Registration of Multiple Point Clouds
KF	Kalman Filter
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KNN	K-Nearest Neighbor
LMS	Least Median Squares
LTS	Least Trimmed Squares
MAE	Mean Absolute Error
MHT	Multiple Hypothesis Tracking
MLP	MultiLayer Perceptron
MSE	Mean Square Error
MV3D	Multi-View 3D networks
NDT	Normal Distribution Transform
PF	Particle Filter
PHD	Probability Hypothesis Density
PIXOR	ORiented 3D object detection from PIXel-wise neural network predictions
PNN	Probabilistic Neural Network
PV-RCNN	PointVoxel-Region-based CNN
RF	Random Forest
RFS	Random Finite Sets

RGB	Red, Green and Blue
RICP	Robust Iterative Closest Point
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROI	Region Of Interest
SD	Standard Deviation
SECOND	Sparsely Embedded CONvolutional Detection
STC	Segmentation-Tracking-Classification
STD	Sparse-To-Dense
SVM	Support Vector Machine
ToF	Time-of-Flight
TP	True Positive
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
V2C	Vehicle to Cloud
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
YOLO	You Only Look Once

Chapter 1. Introduction

1.1 Background

It was reported in 2014 that more people (54 % of the global population) lived in cities than in rural areas (Cohen, 2015). This trend towards city living is predicted to continue, with the urban population expected to make up 66 % of the total by 2050 (United Nations, 2014). However, continuing urbanization is putting significant pressure on city environments including housing, infrastructure, transportation, energy and employment, greatly affecting the quality of life. With an urgent need to alleviate this situation, the concept of 'smart cities' has gained the attention of scientists and practitioners and become a much-studied topic, as it is perceived as a strategy to help mitigate many severe urban problems such as traffic pollution, energy consumption and waste treatment. Smart mobility is one of the six characteristics of smart cities. As one research area in smart mobility, the utilization of Intelligent Transportation Systems (ITS) is foreseen to be beneficial for the economy, the environment, as well as traffic conditions (Benevolo *et al.*, 2016). Some prevailing topics such as Connected Vehicle (CV) are involved in the field of ITS, and traffic monitoring is an effective approach to facilitate the realization of these topics. Among all the sensing technologies, lidar has great potential to be widely employed in traffic monitoring. Related concepts are introduced in sub-sections 1.1.1 to 1.1.4.

1.1.1 Smart cities

The 'Smart City' is generally defined as a concept that involves implementation and deployment of information and communication technology infrastructures to support social and urban growth through improving the economy, citizens' involvement and governmental efficiency (Hollands, 2008). It integrates information and communication technology (ICT), and various physical devices connected to a network (the Internet of things (IoT)) to optimize the efficiency of city operations and services for citizens. In fact, the ideas of smart cities involve the combination of other urban policies such as digital, green, and knowledge cities, therefore, it is a complex, long-term vision of a better urban area (Benevolo *et al.*, 2016).

Smart cities are defined by six characteristics that are described as smart economy, mobility, environment, people, living and governance (Lazaroiu and Roscia, 2012). Each aspect plays a unique role in the construction of a holistic smart city. Smart mobility, as one of the most important facilities to support the functioning of urban areas, involves both environmental and economic aspects and requires both high-tech and virtuous people behaviour (Staricco, 2013; Benevolo *et al.*, 2016). As transportation issues can create severe problems for the quality of urban living, such as pollution and traffic congestion, smart mobility is becoming one of the most important topics in smart city studies. It has the ability to produce significant improvements for the life quality of almost all citizens through objectives that include reducing pollution, relieving traffic congestion, increasing safety, and so on (Frank *et al.*, 2006; Bencardino and Greco, 2014).

1.1.2 ITS

ITS are advanced applications to collect, store and process data, information and knowledge aimed at planning, implementing and evaluating integrated initiatives and policies of smart mobility (Benevolo *et al.*, 2016). ITS incorporate state-of-the-art telecommunication technologies and electronics into transportation systems in order to monitor traffic conditions, appropriately inform drivers, and enhance the efficiency of road networks (Park *et al.*, 2017). Acquisition of important traffic information enables highly flexible monitoring, and the extracted information may be used for various evaluations such as safety (Pyykönen *et al.*, 2010) and emission measurement (Morris *et al.*, 2012). Ultimately, applications of ITS will benefit both the economy and the environment in addition to traffic conditions. It is suggested that, through the utilization of ITS, it is possible to reduce energy consumption by 12%, decrease emissions of pollutants by 10%, increase network capacity by 5-10%, and diminish the number of accidents by 10-15% (Benevolo *et al.*, 2016).

CV, a prevailing topic in the field of ITS, is an advanced sensing and communication technology which enables vehicles, roads and infrastructure to “talk” to each other and share vital transportation information through advanced wireless communication technologies (Zhao, 2019). The communication between vehicles and surroundings can be achieved via Vehicle to Infrastructure (V2I), Vehicle to Vehicle (V2V), Vehicle to Cloud (V2C), Vehicle to Pedestrian (V2P)

and Vehicle to Everything (V2X) protocols. Collisions can be reduced to some extent through adoption of V2V and V2P. Moreover, traffic congestion can be relieved by V2I and V2V communication. However, the full benefits of CV applications can only be realised when all road users are equipped with communication devices. Unfortunately, there are still a certain number of unconnected road users in current traffic flows, which causes a data gap and malfunction of CV systems (Xu *et al.*, 2018). Therefore, one challenge for CV applications is to obtain high resolution microscopic traffic data (HRMTD) of unconnected road users (Wu, 2018b). Unlike macroscopic traffic data which includes traffic flow rates, average speeds, and occupancy, HRMTD refers to the trajectory data of the road users (Xu *et al*. 2018).

1.1.3 Traffic monitoring

HRMTD of road users can be obtained by infrastructure-based traffic monitoring systems relying on different sensors. In the context of traffic monitoring, the acquisition of traffic information mainly presents three challenges: vehicle detection, classification, and tracking (Ambardekar *et al.*, 2014; Park *et al.*, 2017). Vehicle detection plays a vital role in traffic management systems because it can provide important information such as congestion level and statistical analysis of traffic flow (Zhang *et al.*, 2013). Moreover, detection results are regarded as an essential input to traffic monitoring systems and are the basis for subsequent processing tasks such as vehicle tracking (Liu *et al.*, 2013). Vehicle classification is another indispensable aspect since the categories of detected vehicles can supply significant information to ensure that traffic regulations are obeyed, such as certain types of vehicles not appearing in particular restricted areas, or ordinary vehicles not parking in reserved spaces (Ambardekar *et al.*, 2014; Xiao *et al.*, 2016b).

Vehicle tracking is important in urban traffic systems. For traffic condition improvement, tracking vehicles driving through a controlled area can help to observe and hence prevent traffic violations such as speeding, excessive lane changes, as well as drink driving (Sanchez *et al.*, 2010). Moreover, microscopic traffic models operate with detailed and precise traffic data through variables including individual vehicle position, speed, acceleration, and deceleration. All such variables can be acquired through vehicle tracking from various monitoring sensors, with accurate tracking of

individual vehicles necessary in creating HRMTD to serve traffic modelling and emission studies. In addition to the above three components, vehicle reconstruction is also an important aspect in the field of traffic monitoring. On one hand, vehicle size, which can be obtained by measuring the complete vehicle shape, is an indispensable variable for vehicle emission modelling (Pinto *et al.*, 2020). On the other, it would be beneficial for classification and tracking tasks if accurate geometric characteristics of the vehicle could be measured (Xia *et al.*, 2020a; Xia *et al.*, 2020b).

1.1.4 Lidar technologies in traffic monitoring

Previous traffic monitoring systems mainly based on traditional low-cost sensors, and vision-based sensors. With the rapid development in recent years, lidar technologies have shown great potential in traffic monitoring.

Lidar sensors can be divided into two types: flash lidar sensors and rotating lidar sensors. Flash lidar sensors are sometimes referred to as Time-of-Flight (ToF) camera sensors or Time-of-Flight lidar. Since flash lidar only gives very clear resolution in focused areas, rotating lidar is preferred for traffic monitoring from roadside. Existing rotating lidar manufacturers mainly include Velodyne, Robosense, Routescene, Leddartech, Riegl, YellowScan, Quanergy and Geodetics (Zhao et al., 2020).

A lidar sensor is usually composed of a number of vertically configured laser beams covering a wide vertical field-of-view (FOV). Each laser channel is fixed at a specific elevation angle relative to the sensor's central axis (Figure 1.1). These laser beams rotate 360° along the sensor's central axis to form a series of conical surfaces in one scan. Seen as Figure 1.2, two laser beams form their own conical surfaces independently. A panoramic view of the surroundings is thereby recorded in the form of a 3D point cloud. Operating at a high frequency, vehicles can potentially be detected and tracked directly in 3D with high spatial accuracy and temporal resolution. Such panoramic lidar sensors, or some with horizontal FOV smaller than 360°, have been adopted extensively for environment perception in autonomous vehicles (Xiao *et al.*, 2015), however, they are seeing increased use as traffic monitoring sensors due to the ability to capture objects directly in 3D with a high accuracy (Xiao *et al.*, 2016a). The precision of the obtained individual measurements can be as high as 2–3cm. Moreover, with the ongoing development of lidar

technology and increased ubiquity, the cost of such sensors has dramatically decreased in recent years. Therefore, it is foreseen that such sensors will be widely employed in traffic monitoring for smart cities in the near future.

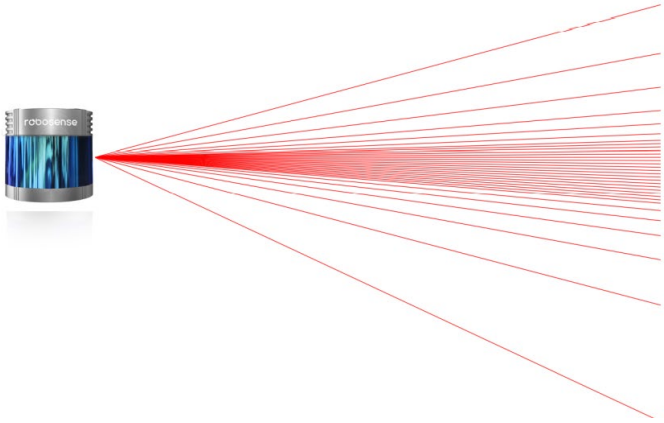


Figure 1.1. Laser distribution of a lidar sensor.

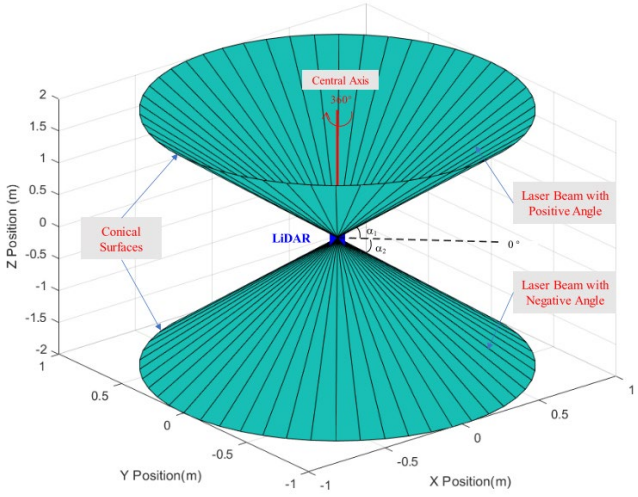


Figure 1.2. Two laser beams rotating around the central axis (Zhao et al., 2020).

1.2 Aims and objectives

From the above introduction, it can be seen that exploiting lidar technologies in traffic monitoring is becoming a popular research area. However, the number of related studies is still small especially those based on roadside lidar system, and the research scope is limited (see summary Section 2.3). Therefore, the research reported in this thesis aims to develop an integrated lidar-

based roadside traffic monitoring system that can provide fundamental traffic information, including the number of vehicles, vehicle dynamics, vehicle dimensions and vehicle types. This overall aim is addressed via the following objectives:

(1) Quantification of vehicle numbers through vehicle detection

Propose a machine learning-based method that can detect vehicles from roadside lidar data with competitive or improved performance with respect to existing methods.

(2) Acquisition of high-quality vehicle trajectories through vehicle tracking.

Develop a vehicle tracking framework with the purpose of increasing the accuracy, improving the completeness and extending the range of the obtained vehicle trajectories. Vehicle dynamics can be inferred from the acquired trajectories.

(3) Measurement of vehicle dimensions through vehicle reconstruction

Based on the tracking results from Objective (2), reconstruct the vehicles to obtain complete vehicle shapes for further analysis.

(4) Identification of vehicle type through fine-grained vehicle classification

Present a method to classify vehicles into one of four fine-grained classes, car, van, bus and truck. This objective is realised as a subsequent procedure of Objective (3), which means the input of the classifier should be the reconstructed vehicles.

1.3 Thesis outline

Following the current introductory chapter, the remainder of the thesis is organized as follows:

Chapter 2 provides a comprehensive literature review of existing traffic monitoring systems and related fields to the proposed system.

Chapter 3 outlines the methodology for each element in the proposed traffic monitoring system.
Chapter 4 firstly reports the results of experiments to evaluate the performance of the proposed traffic monitoring system, then discusses the experiments and the methodologies in order to provide insights on real-world implementation.

Chapter 5 discusses the influential factors and proposes suggestions for real-world lidar installation.

Chapter 6 draws conclusions from the study and recommends future work in the research topic.

Chapter 2. Traffic Monitoring Systems

Considerable advances have been made in traffic monitoring in recent years (Jain *et al.*, 2019). Existing traffic monitoring systems are based on a variety of sensors, such as traditional low-cost sensors, vision-based sensors and lidar sensors. To improve knowledge of current traffic monitoring technologies, existing representative systems are introduced in this chapter. Corresponding studies are summarized in Table 2.1 and explained in sub-sections 2.1 to 2.3. As lidar technologies are developing rapidly in the field of traffic monitoring, there is much research potential in lidar-based monitoring systems. Studies related to each element of an integrated lidar-based traffic monitoring system are summarized in Section 2.4.

2.1 Traditional traffic monitoring sensors

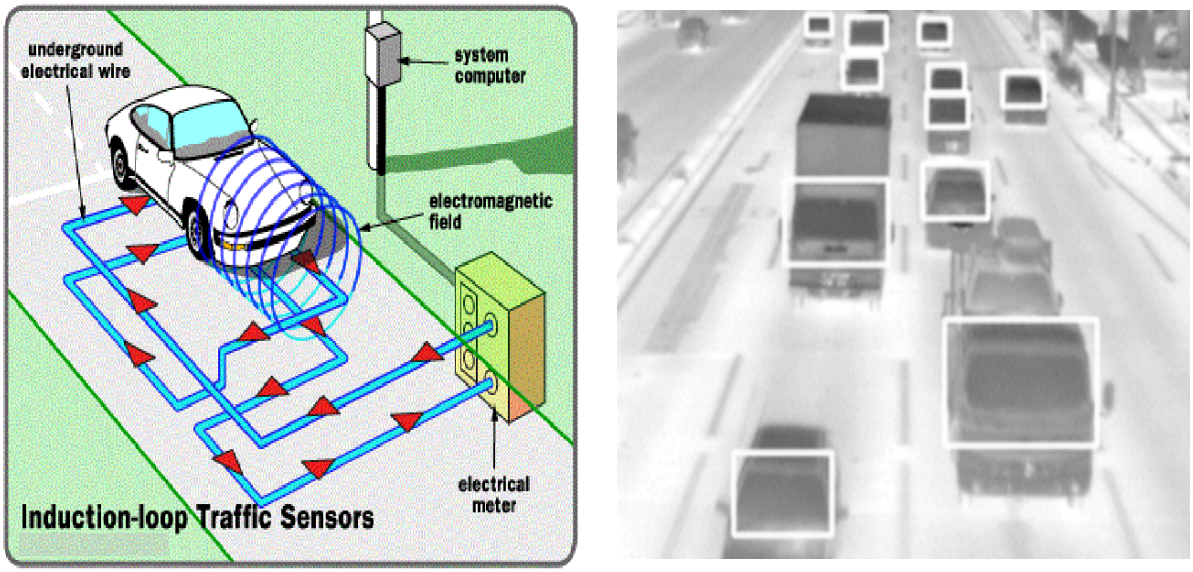
The most commonly used traditional traffic monitoring sensors are inductive loop detectors (Figure 2.1 (a)) and infrared detectors (Figure 2.1(b)). Inductive loop detectors are electrical conducting loops which are insulated and embedded directly in the pavement (Haritha and Kumar, 2017). The oscillation frequency of the inductive loop is directly controlled by the inductance of the loop which changes with vehicle presence. When a vehicle passes over, rests, or stops on the loop area, the inductance of the loop is reduced, showing the presence of a vehicle. The principle is based on a change in the inductance within the loop caused by the metallic components of the passing vehicles (Sulaiman *et al*., 2013; Meta and Cinsdikici, 2010). Loop detectors can be deployed to monitor traffic in individual lanes or across two lanes depending on application. In addition to detecting vehicles, such sensors also can estimate vehicle speeds, as well as classify vehicle types (Lin *et al*., 2004; Ki and Baik, 2006a; Ki and Baik, 2006b; Meta and Cinsdikici, 2010; Ali *et al*., 2011). A set of double-loop detectors, which is known as a speed trap, is commonly used for vehicle speed measurement. However, double-loop speeds that are computed using digital outputs typically have errors between 3% and 5% for ordinary vehicles such as cars and pickups (Ki and Baik, 2006a). Also, inductive loops detector are subject to a high failure rate when installed in poor road surfaces. According to Washington State inductive-loop failure rate survey results which were based on 23 cities and 7 counties within Washington State, the mean failure rate per year was 4.1% among 477 loops (Klein et al. 2006).

The installation of inductive loop detectors decreases pavement life as well as stops traffic during maintenance and repair (Dangi et al., 2012).

Infrared detectors can actively work day and night and obtain information about vehicle position, type, count and speed. The advantages of these infrared detectors are that they can be easily mounted on roadsides and can detect infrared light from large distances over a wide area. Nevertheless, they are very sensitive to extreme weather conditions such as rain, fog and snow (Haritha and Kumar, 2017).

Table 2.1. Summary of existing traffic monitoring Systems

Traffic monitoring Systems		Advantages	Disadvantages	Studies	Output
Traditional sensors	Inductive loop detector	Cost-effective	• Subject to high failure rate when installed in poor road surfaces • Decrease pavement life • Block traffic during maintenance and repair	(Lin *et al.*, 2004; Ki and Baik, 2006a; Ki and Baik, 2006b; Meta and Cinsdikici, 2010; Ali *et al.*, 2011)	Vehicle detection and classification results; Individual vehicle speeds and occupancy time
	Infrared detector	• Easy to install on roadsides • detect infrared light from large distances over a wide area	Very sensitive to extreme weather conditions	(Haritha and Kumar, 2017)	Vehicle position, type, count and speeds
Vision-based	CCTV	• Richer visual information • Modern computer vision technologies to process the information	• Night time detection is difficult • Shadow and vehicle occlusions • Limited field of view	(Im et al., 2016) (Ki *et al.*, 2017) (Bell *et al.*, 2020)	Traffic volume and vehicle speeds
	UAV		Only work continuously for a short time	(Khan et al., 2017) (Puri, 2005)	Traffic volume
Lidar-based		• High fidelity of point cloud measurements • Work day and night • 3D information	• Lack of spectral and textural information • Expensive	(Yao *et al.*, 2012) (Börcs and Benedek, 2013) (Aijazi *et al.*, 2016) (Luo *et al.*, 2016)	Vehicle detection results; Vehicle tracking results.



(a) (b)

Figure 2.1. (a): Inductive loop detectors (Sulaiman et al., 2013); (b): vehicle detection results in infrared Images (Iwasaki et al., 2013).

2.2 Vision-based traffic monitoring approaches

With advanced progress in image processing techniques, vision-based methods have become one of the primary approaches to monitor traffic, with the following advantages: video cameras can produce richer visual information (Figure 2.2(a)) than traditional devices without affecting the integrity of the road and the information can be processed more intuitively with modern computer vision technologies (Tian *et al.*, 2011).

A new method was proposed to automatically calculate traffic volume and vehicle speeds by pattern analysis using pixel data extracted from Closed-Circuit Television (CCTV) video imagery (Im et al., 2016). As the performance of loop detectors to obtain travel speeds is not adequate (see introduction in Section 2.1), a new model was presented for measuring the average link travel speeds using CCTV (Ki et al., 2017). Automatic Number Plate Recognition (ANPR) technology is a mass surveillance method that uses optical character recognition on imagery to read vehicle registration plates (Mathews and Babu, 2017). By using ANPR on CCTV systems, it is possible to monitor individual vehicles, automatically providing information about the speeds and flow of various routes. However, ANPR systems cannot always offer an overall extent of road

users due to specific focus on recognizing characters in car plates and discarding candidate detections if car plates are not fully recognized (Buch et al., 2011). A vehicle detection and tracking workflow via fixed Digital Single-Lens Reflex (DSLR) camera was devised by Bell et al. (2020). Deep learning algorithm You Only Look Once (YOLO) v2 was used for vehicle detection, while Simple Online Real-time Tracking algorithm was enabled for vehicle tracking.

Unmanned Aerial Vehicles (UAVs) are increasingly being used to collect data beyond the range of fixed sensors in order to obtain detailed and accurate data over space and time. This can be particularly useful in areas where fixed sensor infrastructure is not available or it is not financially feasible to install a high density of fixed sensors in an area to be monitored (Khan et al., 2017). Reviews about existing UAV-based traffic monitoring studies can show the advantages of UAVs in an overview perspective. Puri (2005) provided a technical report of research related to the application of UAVs for traffic management. The report pointed out that UAVs can simultaneously view an entire network of roads and inform the base station of emergency or accident sites. Khan *et al.* (2017) presented a universal guiding framework for ensuring a safe and efficient execution of a UAV-based study. The proposed framework provided a comprehensive guideline for an efficient conduction and completion of a drone-based traffic study. It gave an overview of the management in the context of the hardware and the software entities involved in the process. The practical applications of the proposed guiding framework of the UAV-based traffic monitoring and analysis study showed great enlightenment to future research. In addition to review work, concrete examples can provide a better understanding of how UAV can be used in traffic monitoring. A UAV-based smart traffic surveillance system using 5G technology was proposed by Khan *et al.* (2020) (Figure 2.2 (b)) to reduce the number of traffic accidents by managing risks. In the system, the UAV monitors the traffic and detects the excess speed limit and other traffic safety violations on highways and roads over a designated area. When a violation is detected, the UAV warns the vehicle and driver through an integrated module on 1st time as a warning and issue a ticket on the 2nd time and sends the information to the nearby base station. The concerned authorities will take legal action against the violation.

(a) (b)

Figure 2.2. (a) CCTV image data (RAC, 2017); (b) a UVA-based traffic monitoring system(Khan et al., 2020)

Despite the popularity of vision-based methods, it is undeniable that there are issues that make vision-based traffic monitoring challenging. In particular, night time detection is difficult due to poor illumination and sensitivity to light (Robert, 2009). Moreover, for CCTV systems, shadows and vehicle occlusions create difficulty in vehicle detection, tracking and so on. Because of the fixed view angle of cameras, it is often difficult to obtain overall information of vehicles, especially when they undergo sudden motion changes (Saunier and Sayed, 2006). All the above factors affect either the continuity of the monitoring system or the quality of the obtained traffic data. For UAV-based video systems, the biggest issue is that they can only work for a short time due to the limitation of hardware and the flying regulations of the city where the UAV is located, making continuous traffic monitoring infeasible.

2.3 Lidar-based traffic monitoring systems

Recently, lidar technologies have developed rapidly in the field of traffic monitoring due to reduced cost and high fidelity of point cloud measurements (Shirazi and Morris, 2016). Moreover, lidar-based traffic monitoring can work at night because active lidar sensing can be obtained without the requirement of illumination. Furthermore, mobile or fixed lidar systems usually consist of a certain number of laser arrays rotating rapidly around the vertical axis so that the surroundings are continuously scanned, greatly improving the completeness of the acquired data (Xiao et al., 2017). Turning movements of road users at junctions can therefore be monitored,

which is a challenge for traditional traffic monitoring sensors. Another important fact is that lidar data contains 3D information of vehicles that is essential for traffic modelling. Take VLP-16 lidar sensor manufactured by Velodyne as an example. One data frame is generated after the sensor completes a 360° scan and the collected point clouds are stored in a packet capture (pcap) file. For each point, information including the location (X,Y,Z coordinates), the distance to the sensor, intensity, laser ID, azimuth and timestamp is stored in the pcap file. Figure 2.3 shows a frame of raw lidar data collected by a VLP-16 lidar sensor. Different colours of the points represent the intensity of the objects. The main drawback of lidar data is the lack of spectral and textural information. In addition, high profile laser scanners can still be costly. For example, a RoboSense RS-Ruby Laser Rangefinder with 128 beams can cost $30,714.00.

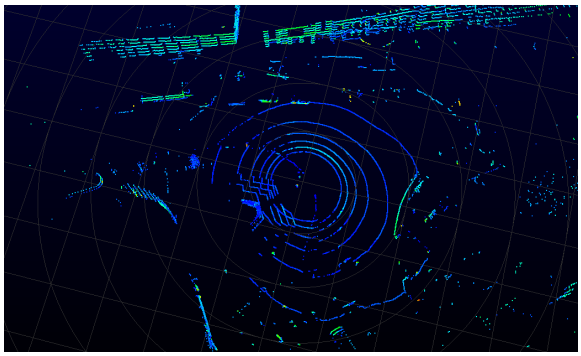


Figure 2.3. A frame of lidar data collected by a VLP-16 lidar sensor.

Yao *et al.* (2012) investigated the theoretical background for airborne laser scanning systems to be used for traffic monitoring. A complete scheme was proposed to analyse urban traffic in real-life situations, which combined vehicle detection successively with the motion classification method. The velocity of the moving vehicle could be derived with knowledge about the vehicle shape. This scheme provided a good representation of the whole traffic situation and the velocity distribution in urban areas. Although the results showed potential in traffic monitoring applications, they were not comparable with those from optical or ground-based sensors. Börcs and Benedek (2012) presented a new model for joint extraction of vehicles and coherent vehicle groups in airborne lidar point clouds collected from crowded urban areas. Firstly, the 3D point set was segmented into terrain, vehicle, roof, vegetation and clutter classes. Then the points with the corresponding class labels and intensity values were projected to the ground plane, where the optimal vehicle and traffic segment configuration was described by a Two-Level Marked Point

Process (L2MPP) model of 2D rectangles. Finally, a stochastic algorithm was utilized to find the optimal configuration. In the proposed model, the vehicles were grouped based on similar orientation, so more complex vehicle arrangement patterns such as strongly curved exit ramps or roundabouts could not be handled. Aijazi et al. (2016) presented a new method for automatic detection of vehicles using a compact 3D Velodyne sensor mounted on traffic signals in the urban environment. The sensor was then mounted on top of a traffic signal to detect vehicles at road intersections. The 3D point cloud obtained from the sensor was first over-segmented into super-voxels and then objects were extracted using a Link-Chain method. The segmented objects were then classified as vehicles or non-vehicles using geometrical models and local descriptors. The results evaluated on real data demonstrated the suitability of the proposed solution for such traffic monitoring applications. However, vehicle and non-vehicle classification stage was still based on a rule-based method while machine learning methods have become more and more widely used. Luo *et al.* (2016) published a novel real-time multiple vehicle detection and tracking system based on a Velodyne HDL-32E sensor. In this system, a radially bounded nearest neighbour algorithm was applied for clustering. Hungarian algorithm procedure and adaptive Kalman filtering were used for data association and tracking algorithm. Even though this article was claimed as vehicle detection and tracking, it was not clearly described how to distinguish vehicles and non-vehicles. The above research has only focused on one or two of vehicle detection and tracking which are only two aspects of an overall traffic monitoring framework.

Despite the fact that extensive studies into traffic monitoring have been conducted, none of the identified traffic monitoring systems have the ability to provide comprehensive traffic information including the number of vehicles, vehicle dynamics, vehicle dimensions and vehicle types simultaneously. As can be seen from the above, vehicle dimensions and vehicle types are largely ignored in most of the approaches. Also, vehicle trajectories from the existing tracking procedures are not of high quality specifically regarding the effective range. Therefore, there is an urgent need to provide an integrated framework for the comprehensive monitoring of traffic.

2.4 Lidar-based traffic information acquisition

As mentioned in the introductory chapter, the acquisition of information in the context of traffic monitoring mainly includes four challenges: vehicle detection, tracking, reconstruction and classification. As can be seen from the review undertaken in Section 2.1, the number of existing lidar-based traffic monitoring studies, especially those covering the aforementioned four challenges, is still quite limited. Therefore, it is necessary to explore the existing lidar-based approaches to each of the challenges, to provide insights into the development of an integrated lidar-based traffic monitoring system. Corresponding literature is summarised in the subsequent sections.

2.4.1 Established vehicle detection methods

An increasing number of methods have been developed for vehicle detection based on lidar systems. These methods can be categorized into either established methods (Section 2.4.1) or state-of-the-art deep learning related methods (Zhang *et al.*, 2020) (Section 2.4.2). Established vehicle detection methods mainly incorporate three stages: background filtering to remove unrelated points; clustering to group points into individual objects; vehicle and non-vehicle classification to distinguish vehicles from other objects (Zhao *et al.*, 2019; Zhang *et al.*, 2020). After background filtering, only moving points from on-road objects such as vehicles and pedestrians remain. These scattered points are then clustered into different groups, with each representing an individual object. The purpose of the following procedure is to distinguish vehicles from other objects, so vehicle and non-vehicle classification is performed by machine learning methods.

2.4.1.1 Background filtering (moving points detection)

In terms of creating HRMTD, foreground information includes all interested road users (e.g., vehicles, pedestrians, and cyclists), while background information ordinarily refers to non-interesting objects such as static background objects (e.g., ground surfaces and buildings) and dynamic background objects (e.g., swaying branches and bushes) (Cheung and Kamath, 2005). This initial background filtering stage not only provides more effective data to track and detect

road users, but also significantly reduces the computational cost in the subsequent procedures (Zhao, 2019).

There are two predominant strategies for background filtering in research: distance comparison (Xiao et al., 2016a; Zhang et al., 2019; Zhao, 2019) and density statistics (Xiao *et al.*, 2017; Wu, 2018a; Wu *et al.*, 2018; Cui *et al.*, 2019). In the first strategy, Zhao (2019) pre-saves the distance information of background objects in a 2D Azimuth-Channel-Distance table (Figure 2.4). Each row of this 2D table represents each azimuth interval of the laser beams during 0° to 360° scan θ; each column of the table indicates each laser beam of the lidar sensor γ and the content of the table is the distance value of the detected background point D, which is shot by a given laser beam at a specific azimuth angle. When a raw lidar frame is input, it is parsed and each point is with attributes of Distance D_i, Orientation θ_i and Laser Beam γ_i. D_i is compared with the distance D from the Azimuth-Channel-Distance table. If the pre-defined filtering criteria is satistified, this point is regarded as a target point and be saved, otherwise it is regarded as a background point and be removed.

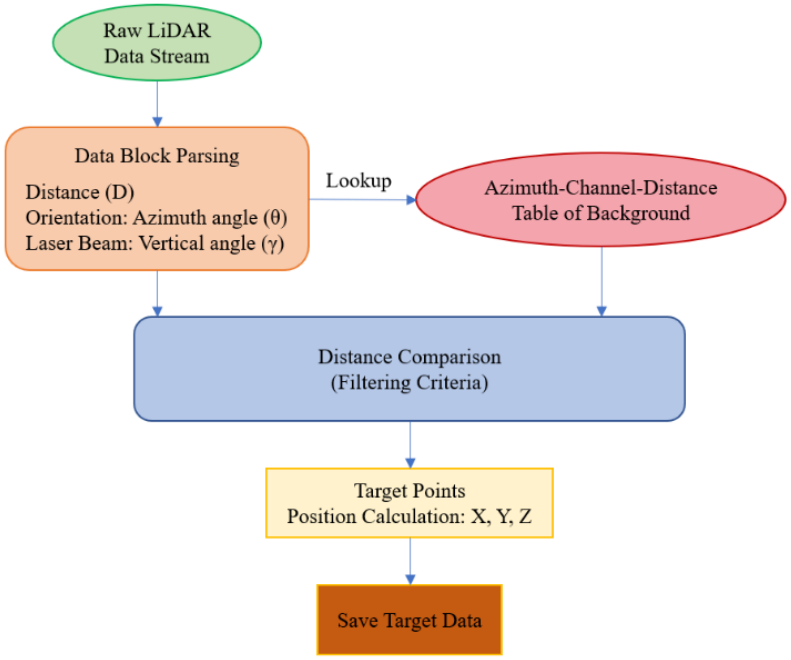


Figure 2.4. Flowchart of Azimuth-Channel-Distance Background Filtering Method (Zhao, 2019).

Similarly, a distance map is used to represent the static background in the proposed Max-distance method from Xiao *et al.* (2016a). Due to the high resolution of lidar data and the complexity of surrounding environments, there may be several distance values for points from the same laser beam at the same horizontal angle in the raw data used for background construction. Therefore, as well as the farthest distance, the mean distance of each laser beam with a certain azimuth angle is also restored in the background dataset by Zhang *et al.* (2019).

In the second strategy, an algorithm referred to as 3D Density-Statistic-Filtering (3D-DSF) is relatively popular in roadside lidar data processing (Wu, 2018a; Wu *et al.*, 2018; Cui *et al.*, 2019), seen as Figure 2.5. The algorithm collects raw data during a certain period as the initial input for background learning. The data frames in a period were aggregated based on lidar point coordinates in Frame Aggregation stage. Afterwards, the 3D space is divided into multiple cubes for the purpose of density statistics, and the point density of each cube is calculated. A threshold of point density of the cubes is then determined to distinguish background and non-background cubes. The background matrix can be constructed once the traverse over all the cubes is finished. When real-time lidar data is input, if the location of a point can be identified in the background matrix, this point is regarded as a background point and removed. Otherwise, the point is considered as a moving object point and kept in the lidar data. Another algorithm, usually called Nearest-Point (Xiao et al., 2017) is introduced from the principle that points are accumulated on static objects, while those on moving objects locate along non-stationary trajectories. Similar to the aforementioned cube, a temporal window is defined, assuming a certain movement speed and a proper object size. The number of neighbour points for the background elements is significantly larger than that of neighbour points for moving objects. One limitation of the second strategy is that the accuracy of the background filtering algorithm directly depends upon the size of each cube or window.

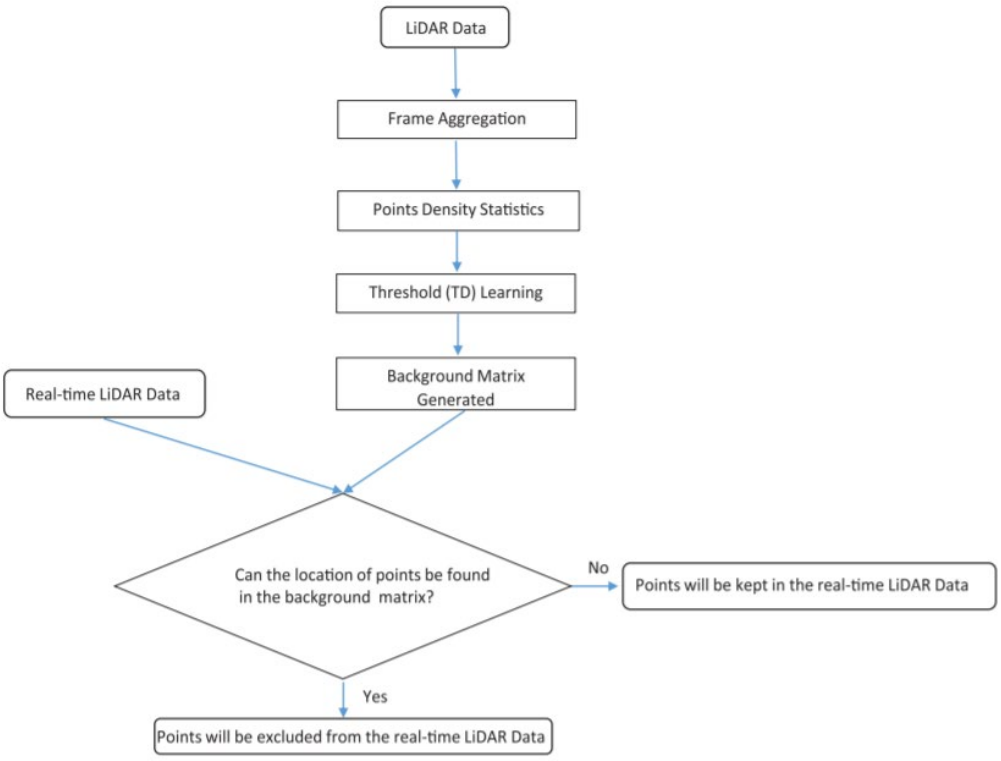


Figure 2.5. Flowchart of 3D-DSF algorithm (Wu et al., 2018).

2.4.1.2 Clustering

In order to cluster objects from lidar data, the related methods can be divided into four categories according to the internal relationships among points: centroid-based; distance-based (originally known as connectivity-based); distribution-based and density-based clustering. The representative method of centroid-based clustering is K-means clustering, in which clusters are represented by a central vector. When the number of clusters is fixed to K, the clustering task is transferred to an optimisation problem: to find the K cluster centres and assign the objects to the nearest one, such that the distance between the object and the selected cluster centre is minimised (Kanungo et al., 2002). The advantage of the K-means clustering method is the low computational cost. However, the number of K clusters must be predefined, which renders the method inconvenient to use in practice. Moreover, since K-means begins with a random choice of cluster centres, varying clustering results may be generated on different runs.

Distance-based clustering methods group objects according to computed distance among points. Thus, the key issue is how to compute distances among points. Except for different options of

distance functions, linkage criteria for clusters such as single-linkage (the minimum object distances), complete-linkage (the maximum object distances), and average-linkage (unweighted/weighted pair group method with arithmetic mean) also needs to be determined. A representative distance-based clustering method is Euclidean Cluster Extraction(ECE) (Dieterle *et al.*, 2017), which is a method for identifying subsets in an input point cloud set, achieved by aggregating points with a recursive nearest neighbour search. The primary parameters consist of the minimum and maximum number of points in one cluster, and the cluster tolerance distance. It is concluded that distance-based clustering methods are computationally expensive for large datasets (Beeferman and Berger, 2000).

The notion of distribution-based clustering methods is that objects within the same cluster have similar distributions. Models that capture correlation and dependence between properties in clusters are needed. One representative method is Expectation-Maximisation (EM) clustering using Gaussian mixture models (GMM), in which the data is modelled with a fixed quantity of Gaussian distributions. The models are initialised randomly, with the corresponding parameters being iteratively optimised to better fit the dataset (Fraley and Raftery, 1998). One disadvantage of distribution-based clustering methods is that it is difficult to ascertain concisely defined mathematical models in practice.

In density-based clustering, points in the higher density areas are grouped as clusters, while points in the lower density areas are considered as either noises or borders. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is the best known density-based clustering method (Ester *et al.*, 1996). It connects points based on the requirement that the number of points within a certain distance threshold (searching radius) satisfy a density criterion (MinPts). Point with more than MinPts neighbours within this radius (including the query point) is considered to be a core point. All neighbours within the searching radius of a core point are considered to be part of the same cluster as the core point (direct density reachable). If any of these neighbours is again a core point, their neighbourhoods are transitively included (density reachable). Non-core points in this set are called border points, and all points within the same set are density connected. Points which are not density reachable from any core point are considered noise and not belong to any cluster. Figure 2.6 illustrates the above concepts. The

MinPts parameter is 4, and the searching radius is indicated by the circles. N is a noise point, A is a core point, and points B and C are border points. The major advantages of DBSCAN include that there is no pre-set for the number of clusters and the clusters can be in arbitrary shapes. However, DBSCAN algorithm fails in case of varying density clusters.

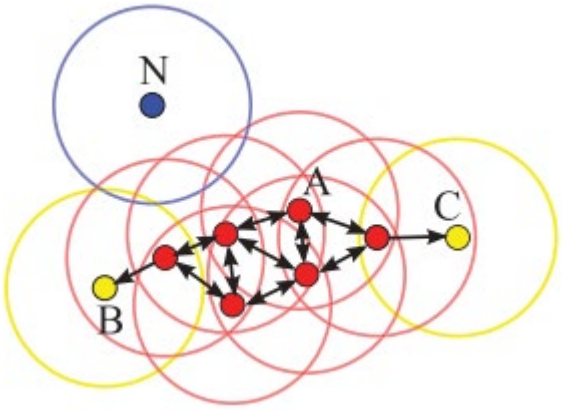


Figure 2.6. Illustration of the DBSCAN algorithm (Schubert et al., 2017).

2.4.1.3 Vehicle and non-vehicle classification

A wealth of studies of object classification from 3D lidar data rely on machine learning strategies. Traditional machine learning has proven to be an efficient approach to object classification from lidar data, with feature selection and classifier training being two important factors in its effective implementation. A summary of some representative methods in recent years has been made in Table 2.2 with regard to the selected features, the adopted classifiers, the data type and the objects classes.

Low-level features based on a small group of points are utilised in the majority of related studies. As shown in Table 2.2, this includes: the number of points in the object cluster (Cui *et al.*, 2019; Zhao *et al.*, 2019; Song *et al.*, 2021; Zhang *et al.*, 2021); object dimension indices including length, width and height (Xiao *et al.*, 2016b; Cui *et al.*, 2019; Zhang *et al.*, 2020; Song *et al.*, 2021); height profile (Cui *et al.*, 2019; Song *et al.*, 2021); difference between height and length (Cui *et al.*, 2019; Song *et al.*, 2021); distance to the lidar instrument (Cui *et al.*, 2019; Zhao *et al.*, 2019; Song *et al.*, 2021; Zhang *et al.*, 2021); and statistics on point distribution in the object cluster (Xiao et al., 2016b; Zhao et al., 2019; Zhang et al., 2020).

As can also be seen from Table 2.2, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), Back Propagation Neural Network (BPNN), Back Propagation Artificial Neural Network (BP-ANN) and Probabilistic Neural Network (PNN) are commonly used classifiers in lidar-based object classification. Some related studies adopted a single classifier (Yao et al., 2011; Wang et al., 2017; Zhao et al., 2019; Zhang et al., 2021), while others experimented on several classifiers and recommended one with the most optimal performance (Xiao *et al.*, 2016b; Cui *et al.*, 2019; Zhang *et al.*, 2020; Song *et al.*, 2021). Naive Bayes, KNN, RF and SVM were tested to detect vehicles from roadside lidar, with RF providing the highest detection accuracy (Cui et al., 2019). SVM, RF, BPNN and PNN were employed to distinguish the road users into ten groups. This demonstrated that SVM utilising the Gaussian kernel function has the greatest ability to classify road-users (Song *et al.*, 2021). SVM and RF were used to recognise vehicles from on-board lidar data, with the two methods showing similar performances. It was noted that RF is capable of evaluating the importance of each feature in the feature set (Xiao et al., 2016b). Similarly, both RF and SVM were assessed with four different feature sets and it was reported that RF performed slightly better than SVM (Zhang et al., 2020).

2.4.2 State-of-the-art vehicle detection methods

There has been an increasing number of deep learning related methods developed for object detection from 3D lidar data in recent years. These methods are summarized as follows:

2.4.2.1 3D object detection with grid-based methods

To tackle the irregular data format of point clouds, most existing works project the point clouds to regular grids to be processed by a 2D or 3D Convolutional Neural Network (CNN). The pioneer work Multi-View 3D networks (MV3D) (Chen *et al.*, 2017) projects the point clouds to 2D birds-eye view grids and places predefined 3D anchors for generating 3D bounding boxes. ORiented 3D object detection from PIXel-wise neural network predictions (PIXOR) (Yang *et al.*, 2018b) projects all points onto a 2D feature map with 3D occupancy and point intensity information to remove the expensive 3D convolutions.

Table 2.2. Vehicle and non-vehicle classification methods

References	Selected features	Classifier	Data	Object classes
(Cui et al., 2019)	Object length Object height Distance between the object and the lidar Number of points Difference between the height and the length Object height profile	Naive Bayes KNN RF SVM	Roadside lidar	Vehicle, non-vehicle
(Song *et al.*, 2021)	Max-length in the trajectory Number of points in the frame with max length Nearest distance from object points to lidar Max-height in the trajectory The difference between length and height Target height profile	SVM RF BP-NN PNN	Roadside lidar	Ten classes including car, bus, pedestrian, et al.
(Wang et al., 2017)	Eigenvalue, eigenvector, histogram of two planes, and slice feature	SVM	On-board lidar	Pedestrian, non-pedestrian
(Yao et al., 2011)	Elongatedness, planarity, vertical position vertical range	SVM	Airborne lidar	Vehicle, non-vehicle
(Zhao et al., 2019)	Total number of points in a cluster The distance of the reference point of each cluster to the lidar sensor Direction of the clustered points' distribution	BP-ANN	Roadside lidar	Vehicle, Pedestrian
(Xiao et al., 2016b)	Object dimension, volumetric feature, relative position, vertical point distribution histogram	RF SVM	On-board lidar	Vehicle, non-vehicle
(Zhang et al., 2020)	The vertical point distribution histogram of the cluster The standard deviation of points in the cluster The volume size of the cluster The area of the 2-D minimum bounding box of the cluster	SVM RF Rule-based	Roadside lidar	Vehicle, non-vehicle
(Zhang et al., 2021)	Number of points (NP) Max intensity change (MIC) Distance between tracking point and lidar (D) Max distance in the XY plane (MDXY) Max distance in Z-axis (MDZ)	PNN	Roadside lidar	Pedestrian, bicycle, passenger car, and truck

Some other works divide the point clouds into 3D voxels to be processed by 3D CNN. For example, VoxelNet (Zhou and Tuzel, 2018) divides a point cloud into equally spaced 3D voxels and transforms a group of points within each voxel into a unified feature representation through the newly introduced voxel feature encoding layer. Graham *et al.*(2018) introduced 3D sparse convolution for efficient 3D voxel processing. Sparsely Embedded CONvolutional Detection (SECOND) (Yan et al., 2018) simplifies VoxelNet and speeds up sparse 3D convolutions. PointPillars (Lang et al., 2019) replaces all voxel computation with a pillar representation, a single tall elongated voxel per map location, to improve backbone efficiency. A two-stage 3D detector, CentrePoint (Yin *et al.*, 2021), uses a standard lidar-based backbone network, i.e. VoxelNet or PointPillars, to build a representation of the input point-cloud. This representation is then flattened into an overhead map-view and a standard image-based keypoint detector is used to find object centres. A summary of the above-mentioned networks is shown in Table 2.3.

Table 2.3. Representative grid-based 3D vehicle detection networks.

Network	Network architecture	Strategy to deal with lidar
MV3D	3D proposal Network+ Region-based Fusion Network	2D bird's eye view grids
PIXOR	Backbone network + Header network	2D feature map with 3D occupancy and point intensity
VoxelNet	Feature learning network+ Convolutional middle layers+ Region proposal network	Equally spaced voxels
SECOND	Voxelwise feature extractor+ Sparse convolutional middle layer+ RPN(Region Proposal Network)	Voxels
PointPillars	Pillar Feature Network +2D CNN Backbone +SSD Detection Head	Pillars
CentrePoint	3D backbone (VoxelNet or pointpillars)+2D CNN+ MLP	Voxels or Pillars

2.4.2.2 3D object detection with point-based methods

Grid-based methods are generally efficient for accurate 3D proposal generation, but the receptive fields are constrained by the kernel size of 2D or 3D convolutions. Therefore, 3D object detection methods operating directly on points have recently emerged.

PointNet is a pioneering effort that directly processes point sets (Qi *et al*., 2017a). The basic idea of PointNet is to learn a spatial encoding of each point and then aggregate all individual point

features to a global point cloud signature. A hierarchical neural network, named as PointNet++ (Qi *et al*., 2017b), was proposed to process a set of points sampled in a metric space in a hierarchical fashion. The set of points are firstly partitioned into overlapping local regions, in which local features are extracted by capturing fine geometric structures from small neighbourhoods. Afterwards, such local features are grouped into larger units and processed to produce higher level features. The above process is repeated until the features of the whole point set are obtained.

Although PointNet and PointNet++ were proposed mainly for scene segmentation, they have served in other novel object detection networks. FPointNet (Qi *et al.*, 2018) proposes to apply PointNet for 3D detection from the cropped point clouds based on the 2D image bounding boxes. PointRCNN (Shi *et al*., 2019a) generates 3D proposals directly from the whole point clouds instead of 2D images for 3D detection only with point clouds, and the following work Sparse-To-Dense 3D Object Detector (STD) (Yang *et al.*, 2019) proposes the sparse to dense strategy for better proposal refinement. Qi *et al.* (2019) propose the VoteNet network which adopts the Hough voting strategy[1] for better object feature grouping. Compared to traditional Hough voting, where the votes (offsets from local key points) are determined by look ups in a pre-computed codebook, Qi *et al*. propose to generate votes with a deep network based voting module. A 3D Single Stage object Detector (3DSSD) introduces Furthest-Point-Sampling based on Feature distance (F-FPS) as a complement of Furthest-Point-Sampling based on 3D Euclidean distance (D-FPS) and develops the first one stage object detector operating on raw point clouds (Yang *et al.*, 2020). The above point-based methods are mostly based on the PointNet series, especially the set abstraction operation (Qi et al., 2017b) which enables flexible receptive fields for point cloud feature learning.

2.4.2.3 3D Object Detection with point-and-voxel methods

As described above, state-of-the-art 3D object detection approaches exploit either 3D voxel CNN with sparse convolution or PointNet-based networks as the backbone. Generally, the 3D voxel

[1] the key concept of Hough voting strategy is to perform a ranking of the image features (such as edges and corners) in the parameter space of the shape to be detected. Votes are counted in an accumulator for which the dimensionality is equal to the number of unknown parameters of the considered shape class.

sparse CNNs are more efficient (Yan et al., 2018; Shi et al., 2020c) and are able to generate high-quality 3D proposals, while the PointNet-based methods can capture more accurate contextual information with flexible receptive fields. If the advantages of two types of networks are integrated, object detection performance can be further improved. According to existing related literature, PointVoxel-Region-based CNN (PV-RCNN) (Shi et al., 2020a) and the improved version of it, PV-RCNN++ (Shi et al., 2021) , are two main networks that have realised the aforementioned integration.

PV-RCNN, a two-stage 3D detection framework, utilizes a 3D voxel CNN with sparse convolution as the backbone for efficient feature encoding and proposal generation. Given each 3D proposal, to effectively pool its corresponding features from the scene, two novel operations are proposed: the voxel-to-key-point scene encoding, which summarizes all the voxels of the overall scene feature volumes into a small number of feature key-points, and the point-to-grid RoI feature abstraction, which effectively aggregates the scene key-point features to RoI grids for proposal confidence prediction and location refinement.

Compared with PV-RCNN, the improvements of PV-RCNN++ mainly lie in two aspects. Firstly, a sectorized proposal-centric key-point sampling strategy, which concentrates the limited key-points to be around 3D proposals to encode more effective features, is proposed for scene encoding and proposal refinement. Meanwhile, the sectorized Furtherest Point Sampling (FPS) is conducted to parallel sample key-points in different sectors to keep the uniformly distributed property of key-points while accelerating the key-point sampling process. Secondly, VectorPool aggregation is proposed as a novel local feature aggregation operation for more effective feature encoding from local neighbourhoods.

2.4.3 Fundamentals of object tracking

Vehicle tracking is a critical element in traffic monitoring systems. Hence, fundamentals including initialization and state update, data association and track management, are introduced in this section to provide the basic knowledge of object tracking. Afterwards, two predominant object tracking strategies, namely tracking-by-detection and tracking-before-detection, are summarized in sub-sections 2.4.4 and 2.4.5.

2.4.3.1 Initialization and state update

A review of related literature shows that Kalman Filter (KF) (Kalman, 1960), Extended Kalman Filter (EKF) (Bar‐Shalom et al., 1990), Unscented Kalman Filter (UKF) (Chen et al., 2018) and Particle Filter (PF) (Doucet et al., 2000) are commonly used filters for initializing and updating tracks in object tracking.

KF is recommended when working with Gaussian noise processes of zero mean in linear dynamic systems and observation models. However, when the system dynamics and observation equations are nonlinear, EKF becomes the better choice. UKF can be used to track both linear and nonlinear motions due to the unscented transform, a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. The above parametric filters are computationally cost-efficient, but they cannot represent complex beliefs caused by data ambiguities. In contrast, non-parametric filters, such as PF, can represent arbitrary beliefs by a set of particles. Nevertheless, the computational cost is at an exponential rate of the dimensionality of the state.

The tracker has to be placed near the measurements in the first place. For first time step, every track is distributed evenly in the tracking region. When the measurements have been received by the next time step, the tracks are relocated to the last known measurement point which has not been associated with the existing tracks. If all measurements have been associated with existing tracks, the remaining uninitialized tracks are to be relocated randomly in the tracking region (Rachman, 2017).

2.4.3.2 Data association

The data association problem in object tracking is classically solved by filtering algorithms such as Multiple Hypothesis Tracking (MHT), Joint Probabilistic Data Association (JPDA), Global Nearest Neighbour (GNN) and Probability Hypothesis Density (PHD).

MHT is one of the earliest proposed multi-object tracking filters. It produces an optimal state estimation by generating a hypothesis (track) for each possible track-measurement pair. The resulting target states from each hypothesis are then estimated using a KF. At the subsequent scan, the new measurements will generate a new set of hypotheses for each track and the

probabilities of these joint hypotheses are updated recursively. The track with the highest likelihood is selected to be the target's state, while all the track hypotheses are maintained. (Reid, 1979; De Feo et al., 1997; Blackman, 2004; Kim et al., 2015).

JPDA was first proposed in the early 1980s. Instead of associating all new measurements to each hypothesis track, it associates the new measurements to existing tracks based on their joint probabilistic score, which measures how well the association between the measurement and the track is. After calculating the score for each possible pair, the tracks are updated with the sum of measurements weighted by their respective scores. However, the traditional JPDA formulation requires that the number of targets/tracks to be known beforehand and remains fixed (De Feo *et al.*, 1997; Rezatofighi *et al.*, 2015).

Both MHT and JPDA suffer from high computational and memory cost. Besides, with an increasing number of targets and measurements, the operation becomes intractable and is not feasible for real time applications. Various work has been done on approximating or simplifying part of the formulation to make both filters tractable at the expense of accuracy (Kim *et al.*, 2015; Rezatofighi *et al.*, 2015).

GNN (Radosavljević, 2006) is another tracking approach, where only a single hypothesis is maintained for each target at each time step. Instead of using a joint probabilistic score for pair evaluation, the best pair is selected to be the one that minimises a particular cost function. The cost function can be in the form of a Mahalanobis distance[2] between each possible pair, a similarity measure in terms of size and shape, or a similarity measure between features of the

measurements to the target. However, it performs poorly when there are crossings among the targets' paths.

PHD shows good robustness and low computational complexity. The targets and measurements are modelled as random finite sets (RFS) instead of dealing with explicit association between

[2] Given a probability distribution Q on R^N, with mean $\vec{\mu} = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^T$ and positive-definite covariance matrix S, the Mahalanobis distance of a point $\vec{x} = (x_1, x_2, x_3, \ldots, x_N)^T$ from Q is : $d_M(\vec{x}, Q) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$

them. A mixture of probability density functions are used to represent the target states. Using RFS allows the filtering problem to be formulated in a Bayesian framework to jointly estimate the number of targets and their states (Mahler, 2003). By using linear Gaussian multi-target model, a closed form solution was able to be formulated by Vo and Ma in 2006. The Gaussian Mixture Probability Hypothesis Density (GMPHD) filter proposed by Vo and Ma has been successfully implemented in numerous real time applications with promising results (Clark *et al.*, 2006; Vo and Ma, 2006; Jeong, 2007; Pham *et al.*, 2007).

As summarized in the above section, traditional filter-based approaches have been widely used in object tracking. However, they are vulnerable to extreme motion conditions, such as sudden braking and turning (Wang et al., 2020b). Besides, those methods tend to struggle if the initial assignment is wrong (Pöschmann *et al.*, 2020).

Deep learning strategies have achieved state-of-the-art results in perception tasks such as image classification, segmentation, and object tracking. Milan *et al.* (2017) proposed the first fully end-to-end multi-object tracking method based on deep learning. The method predicts the assignments of each target, one at a time, using a recurrent neural network (RNN). In contrast, the approach from Baser *et al.* (2019) feeds all detections and their learned similarity scores at once into a CNN to predict the assignments. Besides, Weng *et al.* (2020b) presented two techniques to improve discriminative feature learning for multiple object tracking: (1) instead of obtaining features for each object independently, a novel feature interaction mechanism based on the Graph Neural Network is proposed; (2) instead of obtaining the features from either 2D or 3D space in prior work, a novel joint feature extractor to learn appearance and motion features from 2D and 3D space simultaneously is proposed. Moreover, an end-to-end 3D object detection and tracking network, PointTrackNet, is proposed to generate foreground masks, 3D bounding boxes, and point-wise tracking association displacements for each detected object (Wang et al., 2020b). The network merely takes two adjacent point cloud frames as input and outputs object bounding boxes and corresponding trajectories. A point-wise data association method is designed to reduce the possible negative impacts caused by degraded object detection. Furthermore, a novel optimization-based approach that does not rely on explicit and fixed assignments is proposed (Pöschmann et al., 2020). The result of an off-the-shelf 3D object

detector is represented as GMM, which is incorporated in a factor graph framework. This guarantees the flexibility to assign all detections to all objects simultaneously. As a result, the assignment problem is solved implicitly and jointly with the 3D spatial multi-object state estimation using non-linear least squares optimization.

2.4.3.3 Track management

According to the multi-object tracking literature, the tracking management is mainly concerned with hatching new tracks when objects enter the scene, pruning the tracks during the tracking process and terminating tracks when objects leave the scene.

Normally, all unmatched detections are considered as potential new objects entering the scene (Weng et al., 2020a). According to Weng *et al.* (2020a) and Weng *et al.* (2020b), if a new object is able to find the match in certain frames continuously, it will be assigned an ID and be added to the set of tracked objects. However, if this object stops finding the match before being assigned an ID, the birth count is reset to zero. If a tracked object cannot find the matched detection in certain frames, it is believed that this object has disappeared and will be deleted from the set of tracked objects. Whereas, if this tracked object can still find a match before being deleted, it is considered that the object still exists and the death count will be reset to zero. In the first frame of the data, an empty set is initialized to store the tracked objects.

In order to prevent duplicate tracks associated with the same object, a track pruning mechanism is implemented based on track history or Euclidean distance of the neighbouring tracks (Rachman, 2017). In the first approach, the last n KF states of each track are stored in history in the first place. Then the difference between the states history value of a track towards all other tracks is computed. If the cumulative sum of the standard deviations is smaller than a predefined threshold (history gating level), the track is considered as duplicate. Finally, the track that has the shorter lifetime is deleted (i.e. to preserve track continuity). The second approach computes the Euclidean distance of a track towards each one of others. If the distance is less than the physical distance between two moving traffic objects in practice, the newer track will be deleted. Sualeh and Kim (2019) proposed an example of the second approach: a check is set to ensure that multiple tracks do not get associated with the same object (duplicate tracks) for more than five

consecutive time steps. The states of all tracks are traversed in every time step with Euclidian distance threshold of less than one metre. The track with highest maturity count is retained and the rest are pruned out.

2.4.4 Tracking-by-detection

Most existing lidar-based object tracking methodologies adopt a tracking-by-detection principle in which objects are detected before they are tracked. According to the methodology exploited in object detection, existing tracking-by-detection related studies can be divided into the subsequent two categories.

In the first category (Wu *et al.*; Wu, 2018a; Cui *et al.*, 2019; Zhao *et al.*, 2019; Zhang *et al.*, 2020), object detection is generally realised by the object detection framework introduced in Section 2.4.1 which contains moving points detection (background filitering), clustering and classification. Object tracking is conducted by filtering methods introduced in Section 2.4.3. Several representative studies are summarized as follows:

In the study proposed by Zhao *et al.* (2019), the background filtering algorithm involves frame aggregation, points statistics, threshold learning, and real-time filtering. In the clustering stage, a modified DBSCAN clustering algorithm with adaptive MinPts value and searching radius is developed. After clustering, a reference point is selected to represent each cluster, which will be used in the later procedures. With three hand-crafted features as inputs, a classification model based on BP-ANN is developed to distinguish pedestrians and vehicles in the detection range. A discrete KF is used in the tracking stage.

In the work of Zhang *et al.* (2020), moving points are extracted by Max-Distance algorithm in the first instance. Then they are clustered into individual objects via ECE algorithm. These objects are later classified into vehicles and non-vehicles by traditional classification methods. A tracker composed of UKF and Joint Probabilistic Data Association Filter (JPDAF) is adopted in the subsequent tracking stage.

Wu (2018) developed an automatic 3D-DSF algorithm to filter out the background. A unique operation after background filtering is lane identification which aims to restrict the operation

area to lanes. Consequently, the remaining foreground points only belong to vehicles so that classification is no longer needed. The following procedures mainly include vehicle clustering and vehicle continuous tracking. DBSCAN is adopted in vehicle clustering. To realize vehicle continuous tracking, a tracking point is selected for each vehicle cluster and the GNN algorithm is applied to track the same vehicles in different frames.

In the second strategy (Shi *et al.*, 2019a; Weng and Kitani, 2019a; Weng and Kitani, 2019b; Shi *et al.*, 2020b; Weng *et al.*, 2020a), objects are directly detected from original point clouds by deep learning technologies introduced in Section 2.4.2 and then tracked by a tracker either based on filtering algorithms or deep learning algorithms.

In two typical studies of the second strategy (Weng and Kitani, 2019a; Weng et al., 2020a), two state-of-the-art 3D detectors from Shi *et al.* (2019a), Weng and Kitani (2019b) are experimented with in the detection stage to obtain the bounding boxes. The pre-trained models on the training set of the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) 3D object detection benchmark (Geiger et al., 2013) are adopted. In the tracking stage, a 3D KF predicts the state of associated trajectories from the previous frames to the current frame. Hereafter, a data association module based on Hungarian algorithm (Kuhn, 1955) matches the predicted trajectories from KF and detections in the current frame. Afterwards, KF updates the state of trajectories based on the matched detections. Finally, a module is designed to manage the birth and death of the objects.

The tracker in the above work has been borrowed by Shi *et al.* (2020b) to obtain object IDs of the 3D boxes generated from lidar data by an off-the-shelf 3D object detector, PV-RCNN. Besides, SECOND is used as the 3D object detector in the proposed tracking system considering the detection speed and effect by Wang *et al.* (2020a). 3D KF is used in the following tracking module. Different from the above work in which filtering algorithms are adopted in the tracking stage, deep learning based-method is used for data association by Weng *et al.* (2020b). To be specific, Graph Neural Network has been applied to multi-object tracking for the first time. Meanwhile, a novel feature interaction mechanism is introduced to make the affinity matrix more discriminative.

2.4.5 Tracking-before-detection

Tracking-before-detection is normally adopted to track low-observable objects which are easily overlooked in the traditional tracking-by-detection scheme, or to reduce the complexity or remove the constriants on certain object categories in existing technologies.

As described by Tong *et al.* (2010), making full use of the raw radar data, the tracking-before-detection strategy is suitable for detection and tracking of low-observable objects. A classical PHD filter, with 'standard' multitarget measurement model, is proposed in this work to deal with multi-target tracking-before-detection problem. Besides, an efficient segmentation mask-based tracker which associates pixel-precise masks reported by the segmentation is presented by Ošep *et al.* (2018). This approach utilizes semantic information whenever available for classifying objects at track level, while retaining the capability to track generic unknown objects in the absence of such information. Mitzel and Leibe (2012) proposed a novel tracking-before-detection method that can track both known and unknown object categories in very challenging video sequences of street scenes. Gonzalez *et al.* (2019) raised a track-before-detect framework for multibody motion segmentation based on vehicle monocular vision sensors. The contribution of this work relies on a tightly coupled tracking-before-detection strategy intended to reduce the complexity of existing multibody structure from motion approaches. To remedy fragmented trajectories due to detection failures in the tracking-by-detection framework, a novel detection-by-tracking method that prevents trajectory interruption was proposed by Chen and Tsukada (2019). Based on this method, objects' accurate 3D bounding boxes can be recovered according to the tracking results in the situation of occlusions and missed detections.

The aforementioned object tracking methods based on tracking-by-detection strategy (illustrated in Section 2.4.4) have been confirmed to be efficient in certain aspects. However, they are not qualified to provide more detailed HRMTD due to the negative influence from object detection process. Besides, although tracking-before-detection strategy (illustrated in Section 2.4.5) has great potential to detach tracking from detection, the current small number of approaches for either radar or video sensors cannot be directly applied to lidar sensors. Therefore,

there is still much potential for object tracking from laser scanning systems especially roadside ones.

2.4.6 Vehicle reconstruction

Object shapes from raw point clouds obtained from mobile or roadside laser scanning systems are always incomplete, which results in difficulties in further applications. In the field of traffic monitoring, vehicles are the most concerned investigation targets. Due to their incomplete geometric characteristics obtained from lidar data, it is more difficult to provide high-level traffic data by accurate monitoring of vehicle behaviours. Therefore, reconstructing vehicle shapes from partial lidar inputs shows great importance. The existing shape completion methods can be classified into geometry-based, template-based or learning-based, which are introduced in Sections 2.4.6.1 to 2.4.6.3.

2.4.6.1 Geometry-based shape completion

According to geometric cues (e.g., continuity of local surfaces or volumetric smoothness) of incomplete inputs, geometry-based approaches (Kazhdan *et al.*, 2006; Tagliasacchi *et al.*, 2011; Wu *et al.*, 2015a) can successfully retouch small holes on surfaces of point clouds. When recovering significant missing regions, hand-designed heuristics are applied to complete the 3D shape of objects. For example, Schnabel *et al.* (2009) employed a series of combination of planes and cylinders to guide the 3D shape completion based on partial point clouds. Furthermore, Li *et al.* (2011) proposed an innovative method to learn global relationships between a set of locally fitted primitives. Considering that man-made objects usually have structural regularity, some studies proposed approaches to find regular or periodic structures in geometric models and then use them to complete missing surfaces (Pauly et al., 2008; Zheng et al., 2010). However, these methods heavily rely on the assumption that the input partial point clouds are already of moderate degrees of completion.

2.4.6.2 Template-based shape completion

Another common shape completion strategy is to retrieve a reference from a large-scale database, then to deform or reconstruct the input shape according to the retrieved reference. Pauly *et al.* (2005) produced a complete 3D shape using geometric priors for missing regions from a given 3D shape database, but it requires manual interaction to limit categories of objects. Similarly, Rock *et al.* (2015) explored a method to complete a 3D model of any class automatically from one depth image. However, these methods strongly depend on the capacity of the 3D shape database. To avoid the high dependency of large databases, Shen *et al.* (2012) conducted an assembly approach of geometric primitives to recover 3D structures with a small-scale shape dataset. Sung *et al.* (2015) applied a method to predict the geometric information of an input model and then used a global optimization to reconstruct the entire underlying surface. However, the above methods suffer from several limitations. Firstly, the optimization schemes are too expensive in computational cost for online applications. Secondly, each shape in the pre-prepared 3D shape database requires to be labelled and segmented manually. Finally, these methods are always sensitive to noise.

2.4.6.3 Learning-based shape completion

Recently, exploiting deep learning-based methods for 3D shape completion has become a popular topic. Most of these methods output complete shapes directly from partial inputs using an end-to-end artificial neural network. Wu *et al.* (2015b) constructed a large-scale synthetic object dataset named ModelNet and proposed a Convolutional Deep Belief Networks (CDBNs) to learn shape distributions for completing point clouds. Nguyen *et al.* (2016) integrated CDBNs and Markov Random Fields to recover incomplete shapes. However, these methods all select voxel as 3D data representation since it can be applied in the 3D convolution. Dai *et al.* (2017) explored a 3D Encoder-Predictor Network for estimating a sparse but complete shape, then refined this shape through the nearest-neighbour-based volumetric post-processing. One recent work proposed to directly operate on point clouds for 3D shape recovery (Yuan *et al.*, 2018). Nevertheless, the shape completed using this approach is not uniform, with most of the regions over-concentrated. Also, there is some detailed information lost in the output point clouds.

The above three strategies have been proven to be effective in 3D shape completion. Nevertheless, the geometry-based strategy requires the partial point cloud to be of moderate degree of completion. This requirement cannot always be satisfied. For example, in roadside laser scanning system, the obtained scans cover different parts of the objects and the degrees of completion vary greatly. Besides, a pre-prepared 3D shape database is needed in the template-based strategy. Constructing such a database is both labour and time consuming. Moreover, a large dataset is needed to train the network in learning-based strategy. However, creating such a dataset might be difficult if there is insufficient research data.

In addition to relying on a general template or network as existing strategies do, another attempt is to implement reconstruction merely utilizing lidar data from the target object. As to vehicle reconstruction from roadside lidar, all the clusters of a vehicle are extracted and associated when vehicle detection and tracking are conducted. Each cluster represents an individual part of the vehicle, while successive clusters cover overlapped parts of the vehicle since the scanning frequency of the laser scanner is extremely high. Two successive clusters can be stitched based on the correlation between them. If a series of successive clusters are stitched together, a complete vehicle shape can be obtained. Regarding stitching two or a series of lidar scans, 3D registration and 2D image matching are two predominant methods according to related literature.

2.4.6.4 Pairwise 3D registration (local registration)

The Iterative Closest Point (ICP) algorithm, initially developed by Besl and McKay in 1992 (Besl and McKay, 1992), is often used for 3D point cloud registration. In pairwise registration, the notions ‘fixed’ and ‘moving’ are used to describe the point clouds in a point cloud pair. ’Fixed’, denoted as F, is the point cloud that is considered to have correct coordinate system. ’Moving’, denoted as M, is the point cloud that has to be moved to match the fixed one. The ICP algorithm computes the transformation parameters repetitively by reforming point associations between F and M. The algorithm iterates until one of the following conditions is satisfied: (1) the Mean Square Error (MSE) of the distances between the correspondences is sufficiently small; (2) the MSE difference between two consequent iterations is sufficiently small; (3) the maximum

allowed number of iterations is achieved. The transformation that matches M to F is calculated when the iteration is finished.

ICP only provide optimal local registration results when the initial positions of the overlapping point clouds are close to the registration solution (Brenner, 2009; Shetty, 2017). However, the initial positions of the overlapping point clouds may be far away from each other. Additionally, ICP provides high quality results when all the points or many points in one scan have correspondences in the other (Trucco et al., 1999). While it is quite common in mobile laser scanning data that points in one scan do not have correspondences in the other. Moreover, the execution time of the registration with ICP is considerably high due to nearest neighbour search (Sanchez et al., 2017). To overcome these disadvantages, multiple modified versions of ICP have been developed to deliver improved results.

The Iterative Closest Compatible Point (ICCP) algorithm developed by Godin *et al.* (1994) differs from ICP as it seeks the correspondences between two point clouds under a constraint. The search space is reduced since the corresponding point is searched only among points with similar intensities. Thus, the most computationally expensive operation of ICP, the detection of correspondences, is improved. ICCP, like ICP, performs suitably when most of the points in one point cloud have a correspondence in the other. Another algorithm, the Robust Iterative Closest Point (RICP) (Trucco et al., 1999), applies a Least Median Squares (LMS) method (Rousseeuw, 1984) to eliminate the incorrect correspondences. Therefore, it provides better results than ICP if there is a large number of incorrect correspondences. The Trimmed-ICP algorithm, developed by Chetverikov *et al.* (2002), is based on Least Trimmed Squares (LTS) which was introduced by Rousseeuw in 1984 (Rousseeuw, 1984). This algorithm focuses on the distances between determined corresponding points in point cloud pairs. It indicates that the Trimmed-ICP can handle highly contaminated data. A limitation of this algorithm is that it assumes a fixed overlap of scans (Pomerleau et al., 2013).

The algorithm Iterative Closest Point using Invariant Features (ICPIF) was developed by Sharp *et al.* (2002). It improves the correspondence selection by extracting features invariant to 3D rigid motion from the point clouds. The benefit of ICPIF algorithm is that fewer iterations are needed

than in the ICP to converge to a solution. However, the ICPIF can only construct correct point correspondences when the point clouds are free from noise.

Another widely acknowledged algorithm for 3D point cloud registration is Normal Distribution Transformation (NDT). Instead of using the individual points of the point cloud as in ICP, NDT transforms the set of points residing within a voxel into a normal distribution(Biber and Straßer, 2003). NDT firstly converts the reference point cloud into the normal distribution of multidimensional variables. If the transformation parameters can make the two sets of point clouds match well, the point cloud to be transformed will have high probability density in the reference. As a result, consideration may be given to using an optimized method to work out the transformation parameters that maximize the sum of the probability density, in which case, the two sets of point clouds will match the best (Liu *et al.*, 2021).

The NDT registration algorithm referred to in the study of Liu *et al.* (2021) utilizes the standard optimization technique to determine the optimal match between two point clouds. Since the NDT algorithm does not leverage feature calculation and matching of the corresponding points in the registration process, it is faster than other methods.

As conventional NDT does not generate distributions in cells with the number of points smaller than the threshold, it would fail to represent the environment if the point cloud is divided by high-resolution cells. Also, it can lead to incorrect estimations of pose variations. To solve the problems, a probabilistic NDT representation is proposed by Hong and Lee (2017), in which the probability of a point sample is defined and the mean and covariance are computed based on the probability. The experimental results show that all of the occupied cells have distributions even if the point cloud is divided by high-resolution cells.

To deal with problems of low precision and slow speed when registering large point clouds by existing registration algorithms, Liu *et al.* (2018) propose a new registration method based on feature extraction and matching. The speed of feature point extraction is improved by the judgment of retention points and bumps in the rough registration stage, and the accuracy of the corresponding point pairs is improved by using the random sample consensus algorithm to

eliminate incorrect point pairs. More importantly, an improved NDT algorithm is used in the precise registration phase to further increase the registration accuracy.

In addition to applying individual ICP or NDT algorithm to point cloud registration, efforts have been made to combine them in order to utilize the advantages of each of them. Either the original algorithms or the improved versions are used in the combination.

Registration based on the traditional ICP algorithm is slow, especially when the scale of the point clouds is relatively large. Therefore, Shi *et al.* (2019b) proposes a new registration algorithm in which NDT is used for coarse registration to speed up the process by avoiding using features of the corresponding points to calculate and match. The ICP algorithm is used as fine registration to further improve the accuracy of the overall registration.

A new point cloud registration method based on NDT and improved ICP algorithm is proposed to solve the problem of point cloud registration of laser scanning workpiece position and pose data on industrial pipeline (Xue *et al.*, 2019). Firstly, according to the fast point feature histogram algorithm, the feature points of the point cloud data are extracted to reduce data amount. Then NDT is used to achieve rough registration so that the two point clouds have relatively good initial position and posture. Finally, based on the traditional ICP algorithm, the kd-tree is used to accelerate the searching process of the corresponding point pairs and complete the accurate registration of the point clouds.

In order to meet the needs of intelligent perception of the driving environments, a point cloud registration method based on 3D NDT-ICP algorithm is proposed to improve the modelling accuracy of tunnelling roadway environments (Yang et al., 2021). Firstly, voxel grid filtering method is used to pre-process the point cloud of tunnelling roadways to maintain the overall structure and reduce the data amount. Afterwards, the 3D NDT algorithm is used to solve the coordinate transformation of the point cloud. The cell resolution of the algorithm is optimized according to the environmental features of the tunnelling roadway. Finally, a kd-tree is introduced into the ICP algorithm for point pair search, and the Gauss–Newton method is used to optimize the solution of nonlinear objective function to complete accurate registration.

2.4.6.5 Groupwise 3D registration (global registration)

Groupwise registration refers to the process of aligning all the point cloud scans that have been acquired from laser scanning systems in a common reference system (Nüchter *et al.*, 2005). Traditionally, group registration is performed by repeat pairwise registration among all the laser scans (Evangelidis *et al.*, 2014), such as sequentially strategy (Chen and Medioni, 1992; Blais and Levine, 1995) and one-versus-all strategy (Bergevin et al., 1996; Castellani et al., 2002). The algorithms for pairwise registration, ICP and NDT, are used in the sequentially pairwise registration strategy to update the parameters. These strategies are named as sequential ICP and sequential NDT. The main drawback of sequentially pairwise registration strategy is the error propagation in subsequent steps (Evangelidis et al., 2014; Evangelidis and Horaud, 2017). Simultaneous registration of multiple point sets is another strategy which brings further improvements to the sequential pairwise methods. The cumulative distribution functions Havrda-Charvát (CDF-HC) (Chen et al., 2010), Rényi's second order entropy (Giraldo *et al.*, 2017), t-mixture model (Ravikumar et al., 2018), and Joint Registration of Multiple Point Clouds (JRMPC) (Evangelidis et al., 2014; Evangelidis and Horaud, 2017) are several algorithms involved in this strategy. These algorithms are selected to conduct qualitative and quantitative experiments in the study from Zhu *et al.* (2019b). From the experiments that have been conducted on 2D and 3D data, JRMPC outperforms others in both accuracy and computational complexity.

2.4.6.6 2D image matching

Since the above 3D registration algorithms are computationally complex, especially when the number of laser scans to be registered is large, many researchers attempt to solve the registration problem by using images generated from the point clouds (Christodoulou, 2018). In the method proposed by Lin *et al.* (2017), bearing angle images are generated from the 3D point clouds to highlight the edges formed by angles in diagonal directions, then a feature-based matching method is used to find corresponding pixels between an image pair. Thereafter, 50% of the best corresponding pixel pairs are used, and the corresponding 3D coordinates are tracked back. Lastly, those coordinates are used in least squares approximation to derive the transformation parameters. Initial alignment is not needed in this method and the computation

cost is significantly less than ICP. However, the precision is not better than that of generalised ICP.

Langerwisch and Wagner (2010) presents a novel approach for registering indoor 3D range images using orthogonal virtual 2D scans. The 3D registration process is split into three 2D registration stages such that the computational cost is reduced. Experiments show that this approach is capable of registering 3D range images much more efficient than ICP algorithm. A method developed by Liang *et al.* (2016) retrieves perspective intensity images by applying central projection of the terrestrial lidar data from a viewpoint and employs corner points in the images as tie points to acquire transformation parameters. Thus, intensity is used to reflect the appearance and the geometric structures of objects in order to extract feature points and apply point cloud matching. This method has demonstrated the advantage of using images rather than point clouds for registration by showing the more distinguishable details of object structures.

2.4.7 Vehicle classification

Fine-grained vehicle classification, which refers to detailed categorization of vehicles belonging to the same general class, has been increasingly studied for detailed traffic understanding. Various vehicle classification methods have been developed based on different data sources including videos or images, on-board lidar data and roadside lidar data.

Vision-based vehicle classification has been widely explored. Stark *et al.* (2011) firstly suggest the use of fine-grained category predictions as an input for higher-level reasoning, and secondly design a fine-grained object class representation that captures variations in object shapes and geometries in order to match the object class of interest. Lin *et al.* (2014) propose to optimize 3D model fitting and fine-grained classification jointly.

Deep learning-based vehicle classification has also attracted much attention of researchers in recent years: vehicle classification using an ensemble of local experts and global networks was proposed by Taek Lee and Chung (2017); according to Yu *et al.* (2017), a Faster R-CNN-based model was firstly used to detect vehicles in the existing dataset and then generate images with only one vehicle. Afterwards, a CNN model and joint Bayesian network were exploited to classify

vehicles in a fine-grained way; Ma *et al.* (2019) realised fine-grained vehicle classification with channel max pooling modified CNNs.

In addition to vision-based methods, many studies have also been conducted to use lidar data for vehicle classification. Lee and Coifman (2012) developed a lidar-based classification system that uses data from sensors mounted in a side-fire configuration next to the road. Vehicle shapes were represented with eight features for vehicle classification in this study. Hussain and Moussa (2005) proposed a laser-intensity-based vehicle classification system using a random neural network. The output of this system was one of five major categories: motorcycle, passenger car, pickup or van, single unit truck or bus, and tractor-trailer. Xiao *et al.* (2016b) classified vehicles from on-board lidar data into subcompact (mini or small), compact (hatchback), full-size vehicles (sedan, station wagon, SUV, MPV) and vans. Three sets of features, model, geometric, and the combination, were tested using both SVM and RF classifiers. Wu *et al.* (2019) presented a new approach for vehicle classification using roadside lidar sensor. Six features extracted from the vehicle trajectories were applied to distinguish different classes of vehicles. Naive Bayes, K-nearest neighbour classification, RF, and SVM were applied as the classifiers.

From the above summary of vehicle classification technologies, it can be concluded that vision-based strategies have been well developed with state-of-the-art deep learning approaches. While there is still much potential for vehicle classification based on lidar (especially roadside lidar) data as the amount of related literature is limited.

2.5 Summary.

According to the literature reviews in this chapter, the following research gaps in traffic monitoring are identified:

1. From an overall perspective, there is a lack of integrated traffic monitoring systems that can provide comprehensive traffic information in an end-to-end workflow, thereby determining the number of vehicles, vehicle dynamics, dimensions and types.

2. For vehicle detection, state-of-the-art deep learning methods developed using on-board lidar data have not been applied to roadside lidar data. Although some traditional methods work well on roadside lidar data, there is a trend that they will be replaced by more advanced technologies.

3. For vehicle tracking, the widely adopted tracking-by-detection strategy is not capable of providing high quality HRMTD, especially when detection results are weak. Besides, strategies that attempt to detach tracking from detection (either tracking-before-detection or simultaneous detection and tracking) are still immature, and the number of corresponding studies is quite small, particularly for roadside laser scanning systems. Moreover, due to the incompleteness of the scanned vehicle from the roadside lidar systems, vehicle speeds obtained from tracking procedures are often not accurate enough for HRMTD applications.

4. In terms of vehicle reconstruction, existing strategies rely on a general template or a reconstruction network that needs to be constructed a priori, which might affect the accuracy and efficiency. Hence, it is necessary to develop a method that utilizes only the lidar data of the vehicles to be reconstructed or the results from other elements in the traffic monitoring system.

5. Despite numerous well-developed vision and lidar-based vehicle classification technologies, vehicle classification based on roadside lidar data remains a huge challenge since the data is more sparse and incomplete.

These research gaps are investigated and addressed in this thesis. A roadside lidar-based traffic monitoring system integrating vehicle detection, tracking, reconstruction and classification is proposed in this thesis. For vehicle detection, both traditional machine learning and deep learning methods are proposed to 1) validate the performance of traditional machine learning methods on roadside lidar vehicle detection; 2) explore the potential of deep learning methods on roadside lidar vehicle detection. For vehicle tracking, to improve the accuracy of the obtained vehicle speeds, a vehicle tracking, and high accuracy speed estimation framework is propose; to detach tracking from detection, a joint vehicle detection and tracking framework is proposed. Since clusters of one vehicle has been associated by vehicle tracking, reconstruction of this vehicle is performed on clusters in the near field of the scanning area. Four methods are tested

and the optimal one is selected. Fine-grained vehicle classification is realised via a traditional machine learning method.

The corresponding methodologies are introduced in Chapter 3. Section 3.1 and Section 3.2 are about vehicle detection; Sections 3.3, 3.4, and 3.5 introduce the basic vehicle tracker and two proposed vehicle tracking frameworks; Section 3.6 is about vehicle reconstruction and classification; Section 3.7 is the summary of these methodologies.

Chapter 3. Methodology

The overarching aim of traffic monitoring from roadside laser scanning systems is to derive critical parameters including vehicle numbers, dynamics, dimensions, and types. The first three parameters are acquired in this research by vehicle detection, tracking, and reconstruction, respectively, as shown in Figure 3.1. Due to insufficient training samples, fine-grained vehicle classification, the way to obtain the fourth traffic parameter, is incorporated in the vehicle reconstruction module.

The three-step vehicle detection workflow (Subsection 3.1) is composed of moving point extraction, clustering, as well as vehicle and non-vehicle classification. The third step is realised by traditional machine learning methods. To further improve vehicle detection performance, a 3D object detection network, PV-RCNN, is adopted as a vehicle detector (Subsection 3.2). PV-RCNN is operated on the extracted moving points instead of original lidar data. As can be seen from Figure 3.1, there are two frameworks for vehicle tracking: in the first (Subsection 3.4), a tracking refinement module is designed to increase the accuracy of the estimated vehicle speeds, following an initial tracking procedure based on a tracking-by-detection strategy; in the other (Subsection 3.5), a Joint Detection And Tracking (JDAT) scheme is proposed in order to improve vehicle tracking. Vehicle reconstruction (Subsection 3.6) refers to restoring complete vehicle shapes by aggregating individual vehicle segments from successive frames of roadside lidar data. The complete vehicle shapes with more details are supposed to greatly benefit vehicle type classification in which vehicles are fine-classified into different categories such as cars, vans, trucks and buses.

3.1 Vehicle detection based on a three-step workflow

Vehicles, as the main research targets in this thesis, are expected to be distinguished from the background and other on-road objects. A three-step workflow, shown in Figure 3.2, and a 3D object detection network PV-RCNN, are employed to realize this goal. These two methods are explained in this section and Section 3.2.

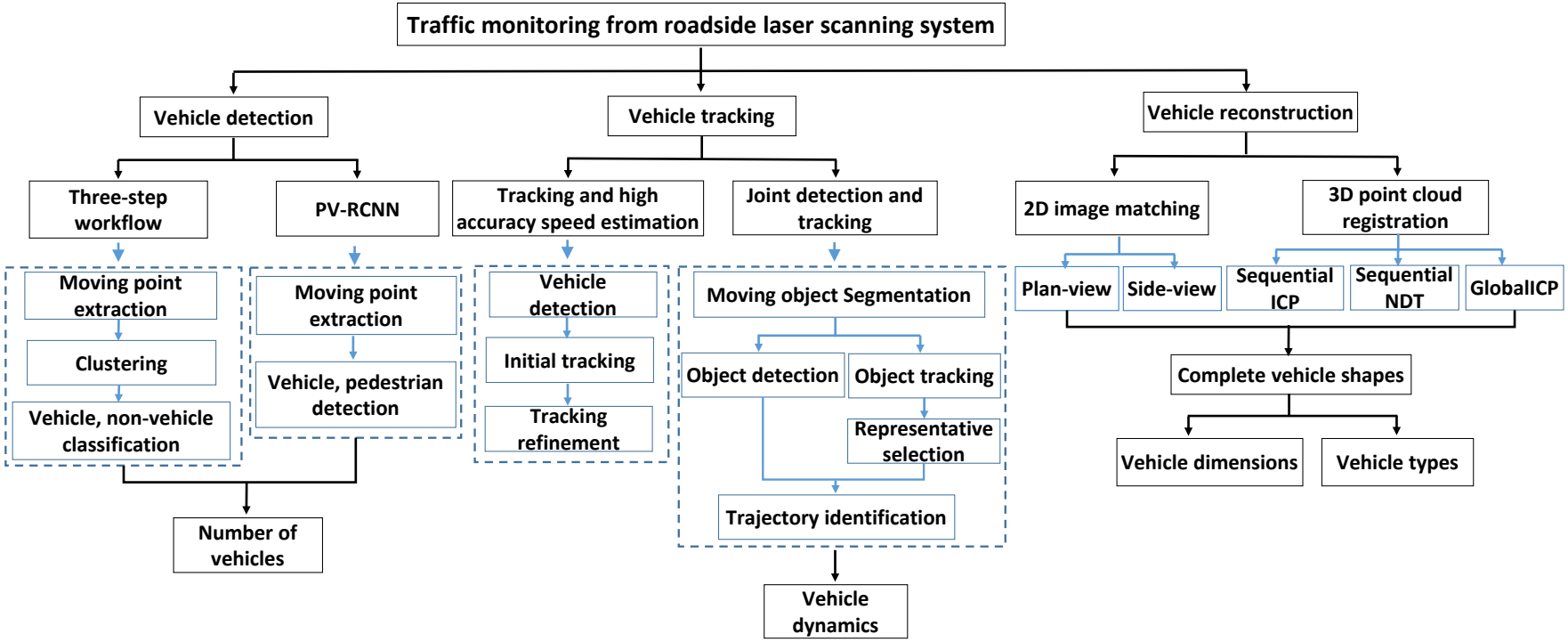


Figure 3.1. Overview of the methodology.

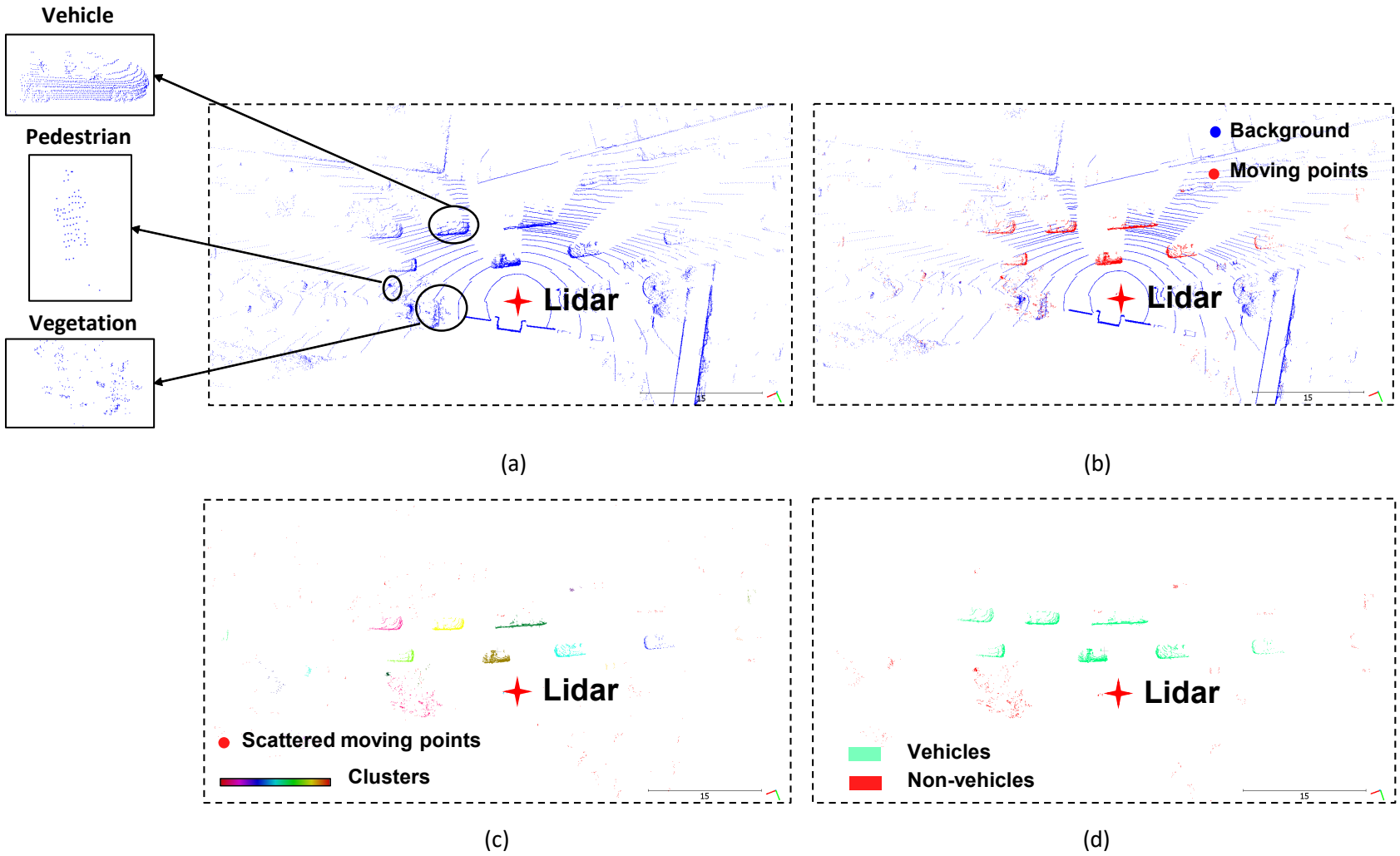


Figure 3.2. Vehicle detection: (a) Original point cloud; (b) Moving point extraction; (c) Clustering; (d) Vehicle and non-vehicle classification.

3.1.1 Moving point extraction

Original point cloud data of urban cities mainly contains moving points on road users or other moving objects, i.e., bushes and trees, and static points on the background, i.e., road and buildings. Since on-road vehicles are the targets of this research, moving points on them are first expected to be distinguished from others. To achieve this goal, the Max-Distance strategy presented by Xiao *et al.* (2016a) is adopted with the following principles: according to the operating mechanisms of the laser scanner, each laser beam rotates in a circle repetatedly with a proper angular resolution (Zhang et al., 2019). A point named as $P_{i\times j}$ is obtained when the i_{th} laser beam interacts with a surface at the azimuth angle j. The distance of this point to the laser scanner can be denoted as $D^{i\times j}$. The furthest point at $(i\ , j)$ with the distance of $D_{max}^{i\times j}$ should locate the static background $(R_b^{i\times j})$, since the laser beam is not supposed to penetrate the background. Meanwhile, if $D^{i\times j} < D_{max}^{i\times j}$, $P_{i\times j}$ is located on a moving object $(R_m^{i\times j})$, as can be seen in equation (3-1). The background of each test site is constructed by determining the furthest point at every location in $R^{i\times j}$. The construction is normally conducted by observations from successive frames within a certain time interval when the number of moving objects is as small as possible.

$$P_{i\times j} \in \begin{cases} R_m^{i\times j}\ ,\ if\ D^{i\times j} < D_{max}^{i\times j} \\ R_b^{i\times j}\ ,\ if\ D^{i\times j} = D_{max}^{i\times j} \end{cases} \quad i \in (1,2,\dots,n), j \in (0, 360^{\circ}) \tag{3-1}$$

3.1.2 Clustering

The extracted moving points should be grouped to obtain individual objects, for which the ECE algorithm is exploited. Two important parameters in the clustering process are the cluster size, *S,* and the minimum distance between two clusters, *d*. The minimum cluster size, S_1, and the maximum cluster size, S_2, should be determined according to the dataset. In terms of *d*, if the value is too small, a real single object can be incorrectly observed as multiple clusters. Conversely, if the value is too large, multiple objects can be regarded as a single cluster. Therefore, heuristic testing on the dataset is required to determine the optimal value of *d*. The ECE algorithm is illustrated in the following steps:

(1) Create a kd-tree representation of the point cloud dataset, P.

(2) Set up an empty list of clusters, C, and a queue of points needing to be processed, Q.

(3) For every point p_i in P, the following operations will be undertaken:

i) Add p_i to Q.

ii) For every point p_k in Q, search the neighbouring points in a sphere with radius r < d. Then check each neighbouring point to see if it has already been processed, if not, add it to Q.

iii) If all points in Q have been processed, add Q to C and reset Q to empty.

(4) Terminate when all the points in P have been processed and included in C.

3.1.3 Vehicle and non-vehicle classification

The purpose of this step is to recognize vehicles from all the moving clusters obtained from the previous two steps. The remaining moving points after background removal belong to either vehicles or non-vehicles. Non-vehicles may include pedestrians, cyclists, motorcyclists and false alarms (e.g., swaying trees and bushes). Therefore, the vehicle detection task is simplified to a binary classification problem that can be realised by traditional machine learning methods. Traditional machine learning methods here refer to supervised learning methods for which feature selection and classifier training are two important factors. Low-level features mainly comprising shape information such as the number of points in the object cluster, object length and height profile, are widely used in vehicle and non-vehicle classification. Commonly used classifiers include SVM, RF, naive Bayes, etc.

3.1.3.1 Feature selection

Inspired by previous studies, the following sub-features are extracted from the object clusters to compose the final feature set: $F = [\, F_1, F_2, F_3, F_4]$.

i) The volume size of the cluster $F_1 = [\, A, L, W, H\,]$.

A: The area of the 2D minimum bounding box of the cluster.

$$L_1 = \max(x) - \min\,(x)$$

$$L_2 = \max(y) - \min(y)$$

$$L = \max\,(L_1, L_2)$$

$$W = \min(L_1, L_2)$$

$$H = \max(z) - \min(z)$$

ii) The standard deviation of points in the cluster: $F_2 = [\, x_s, y_s, z_s]$

iii) The vertical point distribution histogram of the cluster: the proportion of the overall number of points in each vertical section varies among different urban objects (Xiao *et al.*, 2016b). The input cluster is divided into 20 vertical sections from its overall height to the ground:

$$F_3 = [\, h_1, h_2, \dots, h_{20}]$$

iv) Height profile of the cluster

This feature contains the detailed vehicle shape information along the vehicle length (Wu et al., 2019). Each vehicle can be divided into n small equal columns covering the max z value and min z value along the vehicle length direction. In each column, the max height ($MaxH$) can be calculated as $\max(z) - \min(z)$. Then the height profile will be a $1 \times n$ vector with $MaxH$ of each column as sub-features. In the experiment, n is set to 10, according to practice.

$$F_4 = [\, MaxH_1, MaxH_2, \dots, MaxH_n\,]$$

3.1.3.2 The classifier

RF (Breiman, 2001) is adopted as the classifier in this research. RF takes a random number of features to build many decision trees which are assembled and averaged. The optimal parameters, such as number of trees, split quality function and tree depth, are exhaustively searched to acquire the best cross validation results. The handcrafted feature set normally contains a certain number of sub-features. These features are usually selected according to empirical knowledge or other similar works. Whether these sub-features are really helpful and how much they help in the task remains uncertain. One important advantage of RF is that it can evaluate the importance of each feature in the feature set, so that the feature set can be recomposed by features with high importance values. Those with low importance values can be discarded. For comparison, SVM classifier with radial basis function as the kernel function is also implemented as a classifier.

3.2 Vehicle detection based on PV-RCNN

Although traditional classifiers perform well when the clusters of targeted objects are extracted, selecting distinguishable hand-crafted features is a laborious task that somewhat depends on personal experience. Fortunately, the widely and fast developed deep learning technologies provide comprehensive features for the objects through learning mechanisms. As introduced in Chapter 2, PV-RCNN is a recently proposed 3D object detection network that has integrated the advantages of the prevalent point-based methods and voxel-based methods. Besides, PV-RCNN has demonstrated good performance on KITTI data according to Shi *et al.* (2020a). Considering that the difference between data used in KITTI and in this study is primarily the data density, it is anticipated that PV-RCNN will also work well here. Specifically, PV-RCNN is operated on both original lidar data and processed lidar scans containing only moving points.

PV-RCNN is a two-stage 3D detection framework that utilizes a 3D voxel CNN with sparse convolution as the backbone for efficient feature encoding and proposal generation (as shown in Figure 3.3). Given each 3D proposal, to effectively pool its corresponding features from the scene, two novel operations are proposed: voxel-to-key-point scene encoding, which summarizes all the voxels of the overall scene feature volumes into a small number of feature key-points, and point-to-grid Region Of Interest (ROI) feature abstraction, which effectively aggregates the scene key-point features to RoI grids for proposal confidence prediction and location refinement.

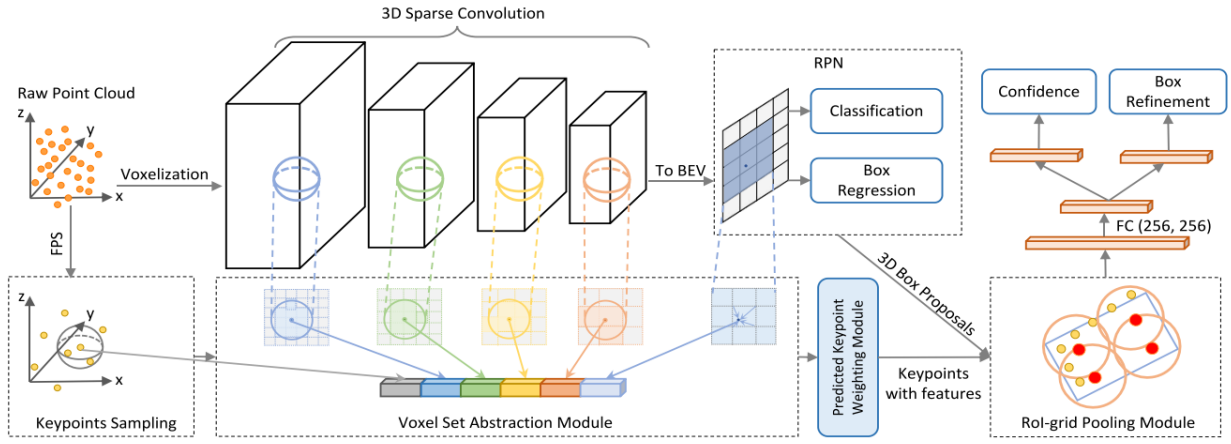


Figure 3.3. The workflow of PV-RCNN (Shi et al., 2020a).

The two operations are described in detail as follows:

3.2.1 Voxel-to-key-point scene encoding via voxel set abstraction

This step aggregates the multi-scale feature voxels into a set of key-points. The multi-scale semantic feature for the key point p_i can be generated by aggregating the above features from different levels of the 3D voxel CNN.

$$f_i^{(pv)} = \left[f_i^{(pv_1)}, f_i^{(pv_2)}, f_i^{(pv_3)}, f_i^{(pv_4)} \right], i = 1, \dots, n \quad (3\text{-}2)$$

Where $f_i^{(pv_1)}, f_i^{(pv_2)}, f_i^{(pv_3)}, f_i^{(pv_4)}$ are the feature vectors from four layers, respectively. Then the key-point features can be extended from the raw point clouds and the bird-view feature maps. Hence, the key-point feature for p_i is further enriched as:

$$f_i^{(p)} = \left[f_i^{(pv)}, f_i^{(raw)}, f_i^{(bev)} \right], i = 1, \dots, n \quad (3\text{-}3)$$

Where $f_i^{(raw)}$ is the raw point-cloud feature, $f_i^{(bev)}$ is the bird-view feature.

3.2.2 Key-point-to-grid RoI feature abstraction

For accurate and robust proposal refinement, 3D proposal (RoI) features are aggregated from the key-point features. Therefore, the key-point-to-grid RoI feature abstraction is proposed based on the set abstraction operation for multi-scale RoI feature encoding. As shown in Figure 3.4, the RoI-grid pooling module is proposed to aggregate the key-point features from the previous step to the RoI-grid points. Each 3D proposal includes 6 × 6 × 6 grid points. The proposal refinement network is able to predict the size and location (i.e., centre, size, and orientation) residuals relative to the input 3D proposal after obtaining each RoI feature. Box refinement can be achieved by the refinement network, which adopts a 2-layer MultiLayer Perceptron (MLP), and the position, orientation, and dimension of cuboid boxes can be obtained.

The PV-RCNN framework is trained end-to-end by the self-created training dataset with the training loss that is the sum of the following three losses: the region proposal loss L_{rpn}, key-point segmentation loss L_{seg} and the proposal refinement loss L_{rcnn}. The three losses are summed with equal loss weights. A Grid search algorithm (Syarif *et al.*, 2016) is adopted in the training process to determine the optimum value for the most important hyperparameters such

as batch-size, epoch and voxel-size.

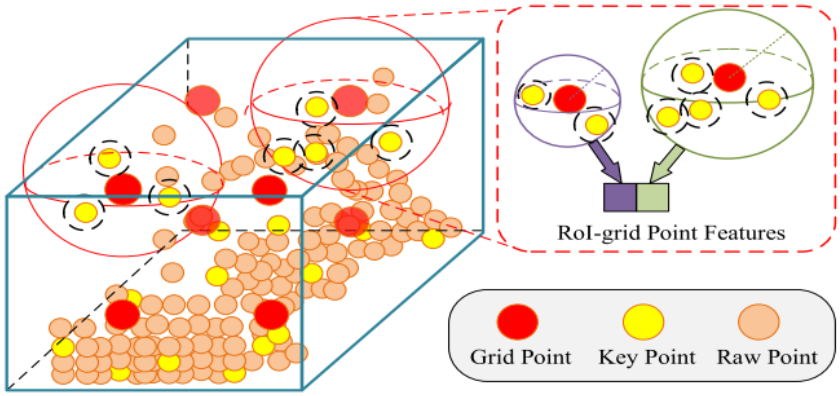


Figure 3.4. RoI-grid pooling module (Shi et al., 2020a).

The original PV-RCNN algorithm was trained by samples of three classes including cars, pedestrians and cyclists from KITTI data. However, in our case, cyclists are not considered as a single class because the number of occurrences in the collected lidar data is extremely small. The two-class training dataset is created using a third-party point cloud labelling software called Supervisely (Deep Systems, 2017). Supervisely is a powerful platform for computer vision development, where individual researchers and large teams can annotate and experiment with datasets and neural networks. It provides a user-friendly interface, a clear documentation and a friendly and reactive support team. For 3D point cloud annotation, the user-friendly navigation in three dimensions makes 3D space annotation easier. Along with additional viewports with top-side-front perspectives using orthographic projections, it gives accurate representation of what you are dealing with. Besides, Supervisely provides more information for accurate labelling and identification with photo and video context. It automatically calculates correlation between 3D space and 2D context and projects the labelled objects on it to achieve unprecedented quality.

3.3 The vehicle tracker

The vehicle tracker in this research is composed of UKF and JPDAF, in which UKF is for initialization and prediction, while JPDAF is for data association. An iteration of the tracking process which includes prediction, data association and state update is introduced in this section (subsections 3.3.1-3.3.3). In addition, as important operations in tracking, track management

development, including the initialization of new tracks, removal of short trajectories and management of occlusions, is also described (subsections 3.3.4-3.3.6).

3.3.1 Initialization and prediction

A constant-velocity UKF is first initialized, which estimates the state of a vehicle by a nonlinear stochastic equation. In constant-velocity motion, the state vector of a vehicle is defined as $x = [x\ ; v_x\ ; y\ ; v_y]$. The state and measurement equations are as follows:

$$x_{k+1} = f(x_k, u_k, t) + w_k$$
$$z_k = h(x_k, t) + v_k \quad (3\text{-}4)$$

where x_k is the state at step k ; f is the state transition function;u_k is the control on the process. The motion may be affected by random noise perturbationsw_k. h is the measurement function that determines the measurements as functions of the state. Typical measurements are position and velocity or some functions of these, which can also include noise represented by v_k.

The prediction is performed using UKF because it performs better than other filters when the predict and update functions are highly nonlinear. The better performance resulted from the usage of the unscented transformation, a deterministic sampling technique to pick a minimal set of sigma points around the mean. Specifically, the sigma points are propagated through system function, f, and the weighted sigma points are recombined into the predicted state and its corresponding covariance. New sigma points are then chosen to be propagated into measurement function h. Finally, the weighted recombination of the sigma points is used to produce the covariance matrix, and the predicted measurement can be directly used to formulate the validation gate. The above steps are thoroughly described by Arya Senna Abdul Rachman (2017).

3.3.2 Data association

JPDAF, a statistical approach, is used to associate measurements to tracks. Instead of choosing the most likely assignment to a target, JPDAF takes the minimum mean square error estimate for the state of each target. At each observation, it maintains its estimate of the target state as the mean and covariance matrix of a multivariate normal distribution.

The measurement validation process is performed to choose associable measurements before passing them to the data association filter. It is realised by setting an elliptical gating area, as shown in Figure 3.5, in which Measurement 1 and Measurement 3 will be discarded because they are considered unlikely to be associated to the predicted track.

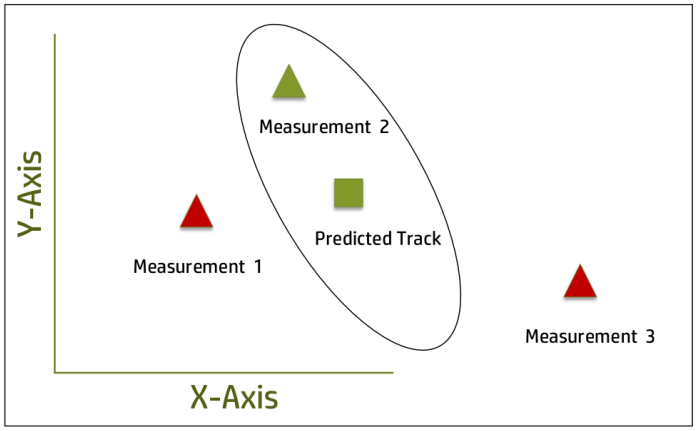


Figure 3.5. Gating process (Arya Senna Abdul Rachman, 2017).

3.3.3 State update

When a new measurement is associated to the track, the state of the track should be updated. Similar to the prediction step, a set of sigma points are to be derived and projected through the observation function h, as shown in Equation 3-5. The sigma points are recombined to produce the predicted measurement and its covariance, and further to generate UKF gain K_k (Arya Senna Abdul Rachman, 2017). The final state update equation is given as Equation 3-6.

$$Z_k^i = h(\chi_{k/k-1}^i) \quad i = 0, \ldots, 2L \tag{3-5}$$

$$\hat{X}_{k/k} = \hat{X}_{k/k-1} + K_k v_k \tag{3-6}$$

3.3.4 Initialization of new tracks

In roadside lidar-based traffic monitoring systems, when a new object enters the scanning region or an object being tracked disappears due to heavy occlusion and reappears after a certain period, it is possible that some measurements could not be assigned to any existing tracks. Therefore, in

such instances a new track should be initiated. In the tracking process, if the association probability of a measurement within the assignment gate is lower than the initialization threshold, a new track will be generated. The association probability of a measurement is given as Equation 3-37 in the work of Arya Senna Abdul Rachman (2017). The initialization threshold is specified as a scalar in the range [0,1].

3.3.5 Removal of short trajectories

The moving targets in this research mainly consist of on-road vehicles, pedestrians, cyclists and motorcyclists. However, there also may be false alarms existing in the extracted moving object clusters, for example from moving leaves on roadside trees or bushes. Through observation, it can be seen that the trajectories of such false alarms are extremely short, since points on such objects typically only move within a limited distance. Therefore, in order to simplify the following procedures, these false alarm trajectories are therefore removed based on their spatial lengths. Specifically, lengths of the extracted trajectories are obtained by calculating the distance between the first and the last cluster, and then compared with a threshold, L . Trajectories with lengths shorter than L are removed.

3.3.6 Management of occlusions

Occlusions are normal issues occurring in traffic data obtained from roadside laser scanning systems. In heavy traffic flow, if the object being tracked is completely occluded and consequently re-observed with an association probability lower than the initialization threshold, a new ID will be assigned to the subsequent detections (see initialization of new tracks).

However, in light occlusion, in which the vehicle being tracked is partially occluded for only a short period, some of the clusters will be incomplete and lower association probability may arise. To ensure continuous tracking in such situations, a small value in the range [0,1] should be assigned to the initialization threshold.

3.4 The vehicle tracking and high accuracy speed estimation framework

As previously mentioned, vehicles can only be partially scanned due to self-occlusion in roadside lidar systems. This has inevitably been responsible for unsatisfactory tracking results, particularly in terms of the accuracy of vehicle speeds, by many current centroid-based tracking methodologies. According to related literature, there has not been any further improvement to solve this issue. A new tracking refinement strategy is therefore proposed and introduced in this section.

As shown in Figure 3.6, the proposed vehicle tracking and speed estimation framework consists of vehicle detection, centroid-based tracking, tracking refinement and vehicle speed validation. Vehicles are detected via a three-step procedure, then tracked by the methodology introduced in Section 3.3. The tracker takes the centroid of the cluster as the vehicle's position, resulting in biases in vehicle speeds due to the incompleteness of the scanned clusters. Accordingly, a tracking refinement module is developed to improve this situation. The core strategy in this module is image matching, in which the vehicle clusters are transformed to 2D images. The estimated speeds are validated against speeds from a reference system mounted on a test vehicle.

3.4.1 Vehicle detection and centroid-based tracking

Vehicle detection is realised by the three-step procedure introduced in Section 3.1. In the third step, traditional machine learning is used to classify vehicles and non-vehicles. The importance of each feature is evaluated by RF, and the full feature set is denoted as $F_{37} = [f_1, f_2, f_3, \dots, f_{37}]$ with the sub-features arranged by the feature importance. In addition to F_{37}, three other feature sets $F_3 = [f_1, f_2, f_3]$, $F_5 = [f_1, f_2, f_3, f_4, f_5]$ and $F_{10} = [f_1, f_2, f_3, \dots, f_{10}]$ are also tested by RF to find the optimal performance for vehicle and non-vehicle classification. After vehicle detection, clusters belonging to the same vehicle in consecutive frames are associated by the tracker described in Section 3.3.

The success achieved in vehicle detection will have direct influence on vehicle tracking. Missed detections will lead to interruptions in the related trajectories, whilst false alarms will create

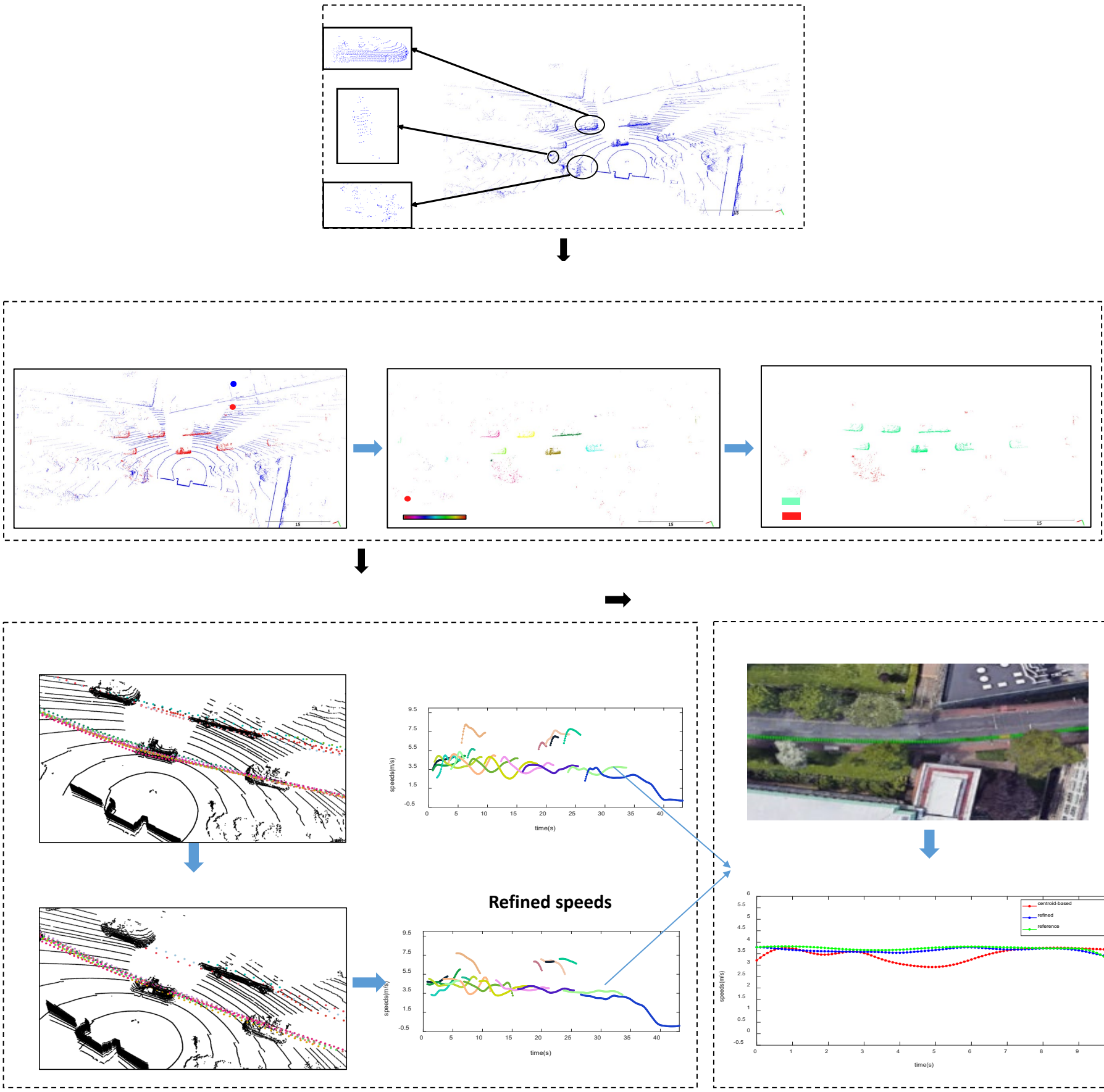


Figure 3.6. Overview of the proposed vehicle tracking and speed estimation framework.

erroneous trajectories in the tracking results. Two parameters in the JPDAF algorithm are related to the first issue: *c*, the threshold for assigning detections to tracks, and *d*, the threshold for track deletion. *c* is usually set to a 1x2 vector [c_1, c_2], where $c_1 \leqslant c_2$. Initially, a coarse estimation is performed to verify which combinations of {detection, track} require an accurate normalized distance calculation. Only combinations for which the coarse normalized distance is lower than c_2 are calculated. Detections can only be assigned to a track if their normalized distance from the track is less than c_1. The values should be increased if there are detections that are not assigned to any tracks and decreased if there are detections that are assigned to wrong tracks. *d* is usually set to [*p*, *r*], where a track will be deleted if it was unassigned at least *p* times in the last *r* updates. Two factors are critical in removing non-vehicle trajectories (either pedestrians crossing the road or other false alarms such as trees): orientation and length of the trajectories. If the orientation of a trajectory deviates too far from others, or its length is too short, the trajectory will be removed.

3.4.2 Tracking refinement

In centroid-based tracking stage, the centroid of the cluster was adopted as the vehicle position. However, the relative position of the centroid changes frame by frame when the vehicle is passing through the roadside lidar sensor. Figure 3.7 (a) shows an example vehicle where F, C and R are the front, centre and rear points, respectively. Figure 3.7 (b) and (c) illustrate the spatial relations between the centroid of the point cloud cluster (C' and C'') and the real centroid C when the vehicle passes the lidar sensor. It can be seen that C' is between F and C when the vehicle is approaching the lidar sensor and the front of the vehicle is scanned; C" is between R and C when it is departing and the rear of the vehicle is scanned. In the time-space diagram of the target vehicle (Figure 3.8), R,F and C are parallel, while the yellow line is not parallel with them as the centroid is closer to F when the vehicle is approaching the lidar sensor and closer to R when it is driven away. The proposed tracking refinement module is intended to rectify the yellow line so as to match the green line to a maximum extent.

After centroid-based tracking, the individual cluster of each vehicle is identified and labelled by the minimum bounding box for subsequent tracking refinement. As to each vehicle ID, the

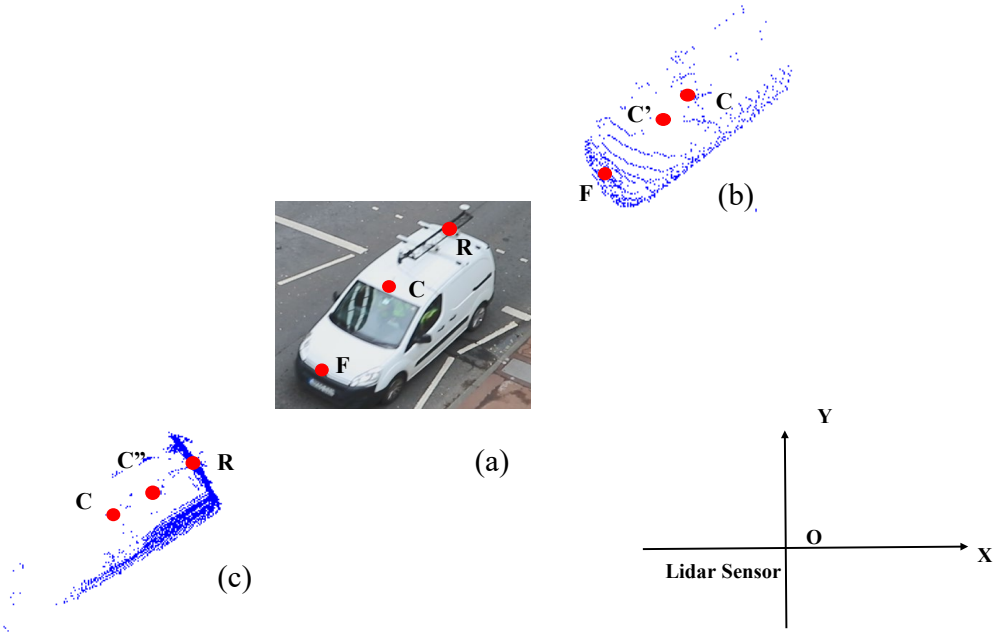


Figure 3.7. A vehicle and key tracking points: F, C, R represent the front, centre and rear, respectively (a) and their spatial relations between the centroid of the point cloud clusters (C' and C") when the vehicle is approaching (b) and leaving (c) the lidar sensor.

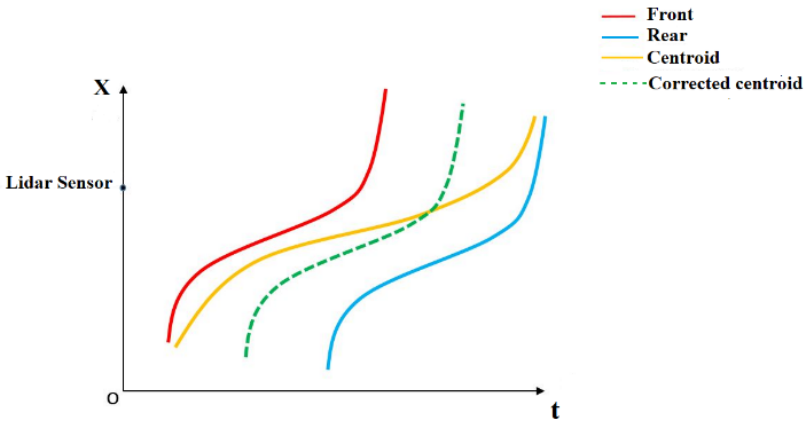


Figure 3.8. Time-space diagram of the target vehicle.

optimized tracking refinement solution relies on determining the correct transformation between two successive clusters. The normal strategy for determining the transformation is frame registration. Enlightened by work from Christodoulou (2018), 3D point clouds can be converted to 2D images to solve the problem by image matching. It is noteworthy that the conversion here is implemented on the previously extracted 3D vehicle clusters rather than the entire lidar frame. The process of the proposed tracking refinement comprises three steps:

conversion from 3D cluster to 2D image, image matching, and 2D to 3D transformation. Tracking refinement is performed within pairs of successive vehicle clusters in the plan view. Figure 3.9 displays the process for one example pair, which is described in the following subsections.

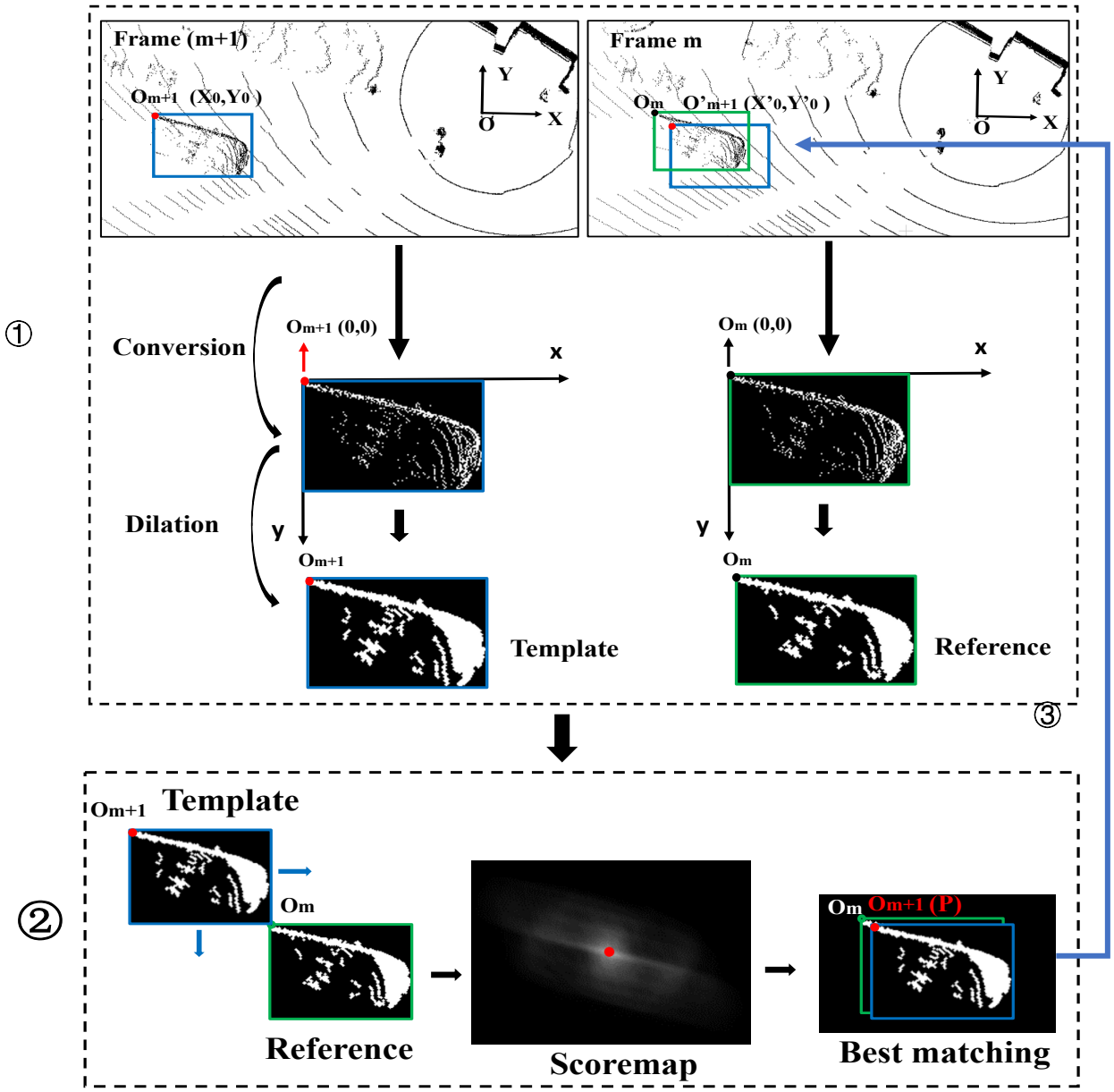


Figure 3.9. The proposed tracking refinement module.

3.4.2.1 Conversion from 3D cluster to 2D image

As shown in Figure 3.9, Frame m and Frame $(m+1)$ are projected to 2D in the plan view in the first instance, and all the points are in the laser scanner coordinate system XOY. Under this

condition, $O_{m+1}(X_0, Y_0)$ is the origin of the minimum bounding box around the vehicle cluster in Frame $(m+1)$. Correspondingly, $O_{m+1}(0,0)$ is the origin of the image, which is located in the image coordinate system xoy . Equation 3-7 shows the conversion from a point on the vehicle cluster to a pixel in the corresponding image, where $pixelsize$ refers to the resolution of the image.

$$x = (X - X_0)/pixelsize$$
$$y = -(Y - Y_0)/pizelsize \tag{3-7}$$

The parameter $pixelsize$ can be decided by testing plausible values in the experiment. In our situation, two case studies in which a test vehicle was tracked have been used to determine the optimal value in a range from 1cm to 10cm (values outside of this range are considered either too small or too large according to the data density). RMSE between the estimated speeds and the reference is calculated for each pixel size. Based on the test, a pixel size of 3cm generated the lowest RMSE and was therefore chosen as the optimal value.

3.4.2.2 Image matching

Image matching is intended to determine the optimum location of a template within a reference image. The image generated from Frame $m+1$ is regarded as the template, whereas Frame m is the reference. The template image shifts pixelwise over every possible location in the reference image. Based on the cross-correlation coefficient metric, a similarity score $S(x, y)$ is calculated between the template and the corresponding sub-image in the reference, accordingly (see Equation 3-8). A score map with each pixel assigned a similarity value is formed after completion of the search process. The optimum matching location, namely the lightest point (red dot in Figure 3.9) in the scoremap, is where the largest score is determined (See Equation 3-9).

$$S(x,y) = \frac{\sum_{i=1}^{m_1}\sum_{j=1}^{n_1} T(i,j)R(y+i-1, x+j-1)}{\left[\sum_{i=1}^{m_1}\sum_{j=1}^{n_1} T(i,j)\right]^{1/2}\left[\sum_{i=1}^{m_1}\sum_{j=1}^{n_1} R(y+i-1, x+j-1)\right]^{1/2}} \tag{3-8}$$

$$P = \arg\max_{x,y}(S(x,y)), (x = 1,...,n_2; y = 1,...,m_2) \quad (3\text{-}9)$$

In Equation 3-8, T is the template image with (m_1, n_1) pixels; R is the reference with (m_2, n_2) pixels; (x, y) is the origin of the sub-image corresponding to T in R. In Equation 3-9, P is the optimal matching location in R.

3.4.2.3 2D to 3D transformation

Real-world coordinates are reserved for each pixel according to Equation 3-7. Consequently, P can be located on Frame m, labelled as $O'_{m+1}(X'_0, Y'_0)$ in Figure 3.9. Considering its position $O_{m+1}(X_0, Y_0)$ in Frame $m+1$, the displacement can be calculated and the vehicle speed obtained. The speed values during the entire tracking period will be estimated when a chain of the above operations has been fulfilled amongst all the tracked clusters of the same ID. A Gaussian window with size s=20 is used to smooth the acquired values so as to filter out noise.

3.4.3 Vehicle speed validation

In this section, the concept of the speed reference system is introduced and the calculation process of the reference speeds is explained.

A test vehicle equipped with an independent speed reference system is used to validate estimated vehicle speeds. The reference system is composed of two Global Navigation Satellite System (GNSS) antennas, an Inertial Measurement Unit (IMU) unit and an odometer (seen as Figure 3.10). The GNSS and IMU unit are mounted on top of the vehicle, while the odometer is installed on one of the rear wheels to improve positional accuracy. The test vehicle is driven through the scanning area for several rounds at the test sites.

The following operations are conducted to obtain the reference vehicle speeds: original vehicle position acquisition, post-processed vehicle position acquisition and vehicle speed calculation. The original data (.lpd) from the reference system is first parsed to GPS data (.gps) and lidar data (.pcap). The GPS data, renamed as .anpp from .gps, is then imported into Spatial Dual Manager (Advanced Navigation, 2020) to obtain the original vehicle positions (red trajectory in Figure 3.11).

The .anpp file is further processed using Kinematica (Advanced Navigation, 2021) to acquire more accurate vehicle positions (blue trajectory in Figure 3.11)).

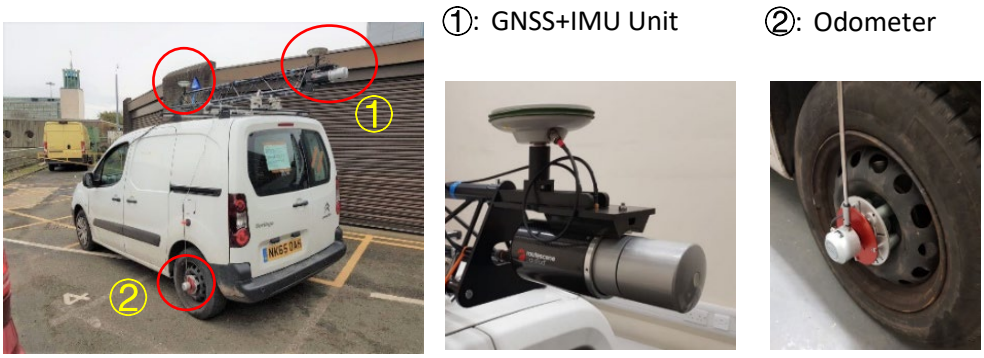


Figure 3.10. Installation of the speed reference system.

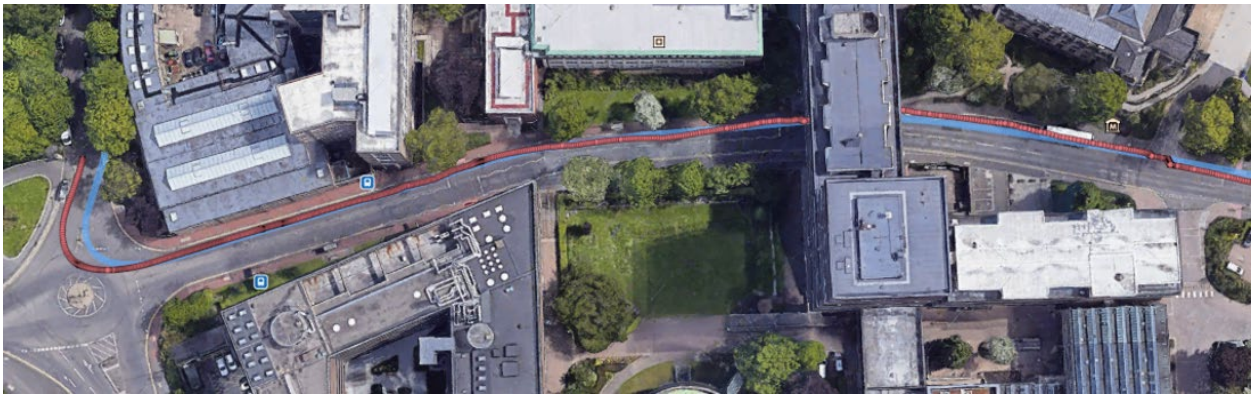


Figure 3.11. Trajectories before (Red) and after (Blue) post-processing

Since vehicle coordinates in the post-processed data are in the ETRS89 (European Terrestrial Reference system 1989) coordinate system, Grid InQuestII software (Ordnance Survey, 2016) is used to perform an accurate conversion from (Latitude, longitude, Height) to (Eastings, Northings, Height) in the OSGB36 National Grid coordinate system.

3.5. The JDAT framework

The tracking-by-detection strategy illustrated in Section 3.4 has achieved promising performance in tracking all the detected objects. However, the resulting object trajectories might be shortened or discontinuous due to the fact that this strategy is sensitive to miss detections . To address the issue, a JDAT scheme, where object tracking is independent from object detection, is proposed in this research (Figure 3.12).

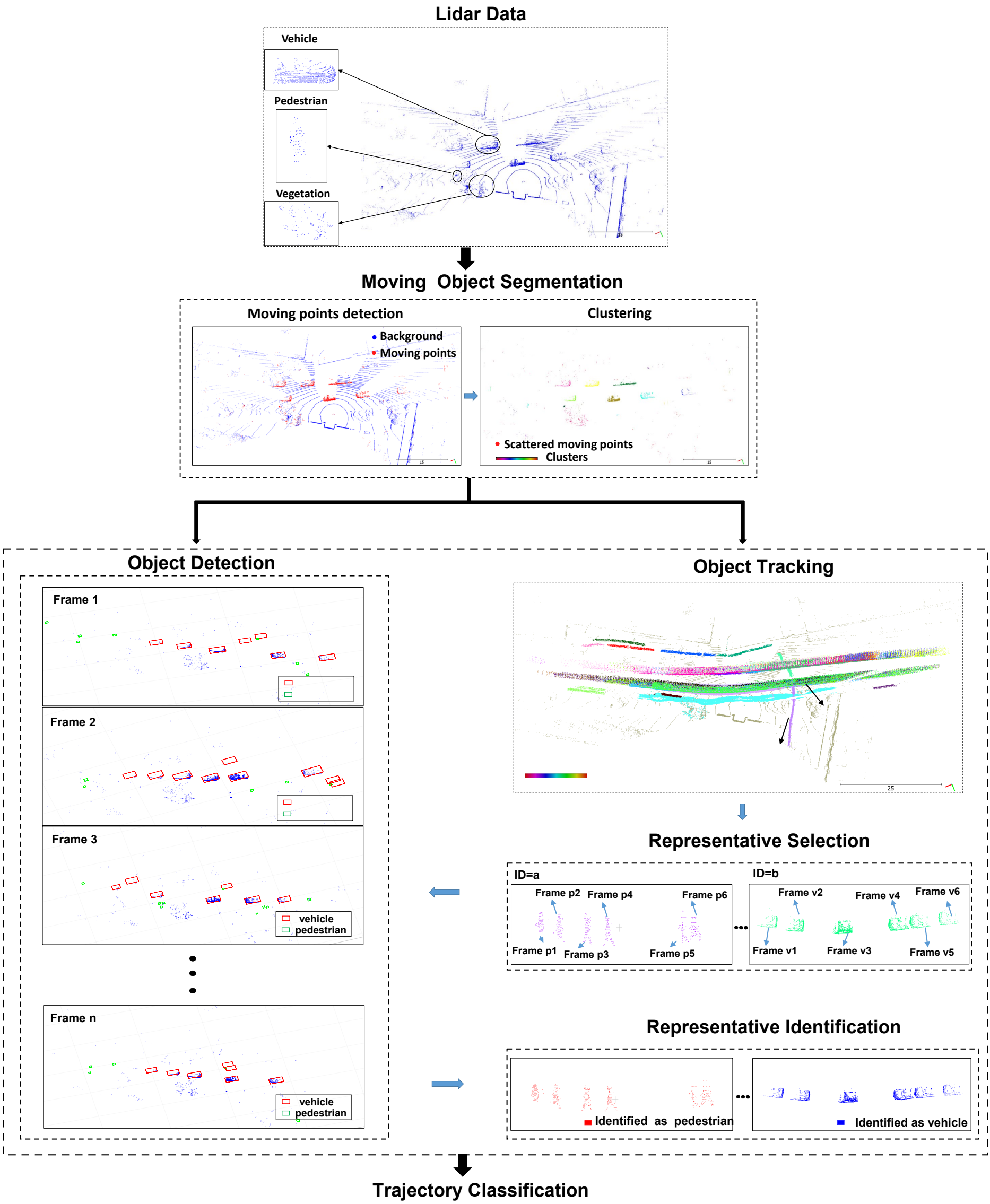


Figure 3.12. Flowchart of JDAT

The moving objects are segmented firstly by moving point extraction and then instance clustering. Afterwards, two procedures, object detection and tracking, are conducted in parallel. In object detection, PV-RCNN is employed to detect vehicles and pedestrians from the extracted moving objects. In object tracking, the tracker introduced in Section 3.3 is used to obtain the trajectories of all the moving objects. A certain number of representatives for each trajectory are then selected and the category of each representive can be determined from the results of the object detection procedure. Subsequently, the identities of the trajectories are determined.

Compared with the vehicle tracking framework in Section 3.4, improvements of JDAT in this section mainly rely on the following aspects:

Firstly, instead of adopting a tracking-by-detection strategy, as in the first framework, tracking starts immediately after moving object segmentation and in parallel with object classification, which guarantees all observable clusters can be associated to corresponding tracks. It should be noted that achievieving maximum-range tracking is one of the aims of this framework, so small object clusters with a few number of points should be maintained, which means the parameter minNumber in the clustering stage should be as small as possible.

Secondly, only clusters that are more distinguishable than others, namely the representatives, participate in trajectory classification. Two important processes, selection and identification of representatives; determination of trajectory categories, are elaborated as follows:

3.5.1 Selection and identification of representatives

After tracking, clusters of a single object have been associated across successive frames. However, not all of them are needed in the cassification process because they belong to the same category. Since clusters with larger sizes are more visible and distinguishable than those with smaller sizes on a trajectory, they can act as representatives of the trajectory to be fed into the classifier such that the negative influence from the low-observable clusters can be minimised.

The important operation after Representative Selection is to identify the categories of the representatives according to the results from object detection, so that the category of the corresponding trajectory can be determined.

One attribute of the representatives is the ID of the original lidar frame where the representative is extracted. As described in Section 3.2, PV-RCNN in object detection is operated on frames that only contain moving points abstracted from the corresponding original lidar frame, so the category of the representatives can be easily traced from the detection results by their frame IDs.

3.5.2 Determination of trajectory categories

The number of representatives for an individual trajectory is n. If at least p (a ratio) of the representatives are classified as one of the classes in the detection results, the trajectory is classified into that class. n and p should be decided by the datasets and the performance of the classifier, which is discussed in Chapter 4.

3.6 Vehicle reconstruction

All clusters of a vehicle have been associated across successive frames after the tracking process. Each cluster represents an individual part of the vehicle according to the roadside laser scanner's perspective. Therefore, in theory, if all vehicle portions are stacked together according to the transformations within the pairings, a more complete vehicle shape can be constructed. In practice, to reduce error accumulation and computing cost, vehicle reconstruction is only conducted in near field of the scanning range, which means only vehicle clusters with distance to the laser scanner smaller than a defined threshold will be considered in the reconstruction. To realize vehicle reconstruction, 2D image matching and 3D point cloud registration strategies are adopted in this research.

3.6.1 Vehicle reconstruction by 2D image matching

As in the tracking refinement module in Section 3.4.2, two successive vehicle clusters from Frame m and Frame $(m+1)$ can be converted to a pair of plan-view 2D images. In the following matching process (see Figure 3.13), the second image is regarded as the reference R, and the other is the template T. With the best matching position of T in R found as the brightest point in the score-map in Figure 3.13 (also P in Equation 3-9), the transformation between two images is determined, given as Equation 3-10. With regard to vehicle reconstruction, a series of similar

image pairs are needed. If the first of these images is taken as the reference, all the others can be transferred to it according to the transformations, seen as Equation 3-11.

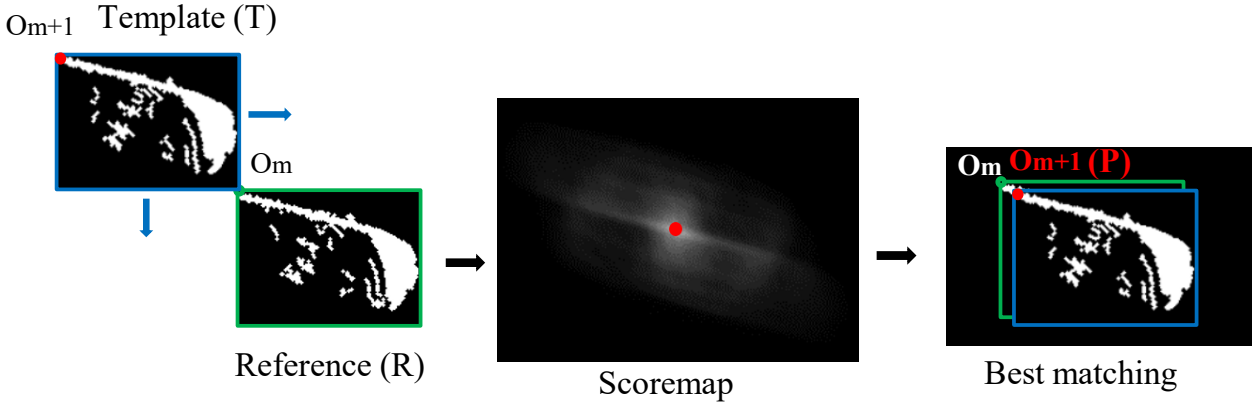


Figure 3.13. Matching process between two images.

$$T_{m+1,m}(x,y) = (P_x, P_y) \tag{3-10}$$

$$T_{n,1}(x,y) = \sum_{m=1}^{n-1} T_{m+1,m}(x,y) \tag{3-11}$$

3.6.2 Vehicle reconstruction by 3D point cloud registration

Vehicle shape reconstruction is essentially about merging a set of successive point clusters which represent different parts of the vehicle based on the transformations among them. Frame registration is the basic way to calculate those transformation parameters. Observed from the existing literature about multiple point cloud registration, sequentially pairwise registration and simultaneous groupwise registration are two predominant strategies. ICP and NDT are considered as two potential algorithms for registration within pairings. NDT is shown to converge from a larger range of initial pose estimates than ICP, and to perform faster. However, the poses from which NDT converged are not as predictable as ICP. Considering that there might be error accumulation in sequentially pairwise registration strategy, simultaneous groupwise registration in which the alignment of multiple point clouds is refined is regarded as a better option. GlobalICP is a predominant simultaneous groupwise registration algorithm. In order to make a thorough comparison, two strategies, including three algorithms, are experimented with in this study.

3.6.2.1 Sequentially pairwise registration

The idea of sequentially pairwise registration is to determine the transformation between each successive pair of point cloud scans, and transfer each scan to the coordinate system of the reference, which is usually the first scan by multiplying the transformations of scans in-between them. Sequential ICP and sequential NDT are two typical algorithms in this strategy, of which the basic algorithms are ICP and NDT, respectively. ICP is a well-known method in the field of point set registration for rigid transformation between two point clouds. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize an error metric such as the sum of squared differences between the coordinates of the matched pairs. ICP can be expressed as an optimization problem:

$$\arg\min_{R,t}\left\{\frac{1}{M}\sum_{j=1}^{M}\left\|y_j-(Rx_j+t)\right\|_2\right\} \tag{3-12}$$

Where $(x_i\ , y_i\ )$ is a correspondence pair; R is the rotation matrix and t is the translation vector of a correspondence pair; M is the number of correspondence pairs.

NDT is a registration algorithm that is applied to the statistical model of 3D points and uses standard optimization techniques to determine the optimal match between two point clouds. Its main idea consists of modelling point clouds with a set of normal distributions generated by discretising each point set into voxels (Slama, 2017). Pairwise point cloud registration based on ICP and NDT are illustrated in Figure 3.14.

If there are a series of successive point cloud scans to be registered by sequentially pairwise registration, each scan should be transformed to the reference scan (usually the first scan) based on Equations 3-13 and 3-14. $T_{i,i-1}$ is the transformation between the i_{th} scan and the $(i-1)_{th}$ scan, as denoted in Equation 3-13, in which R represents the rotation and t denotes the translation of two successive scans. The transformation of the j_{th} scan and the reference scan, $T_{j,1}$, is shown in Equation 3-14. Sequentially pairwise registration is realised when all the scans are transformed to the reference scan according to Equation 3-14.

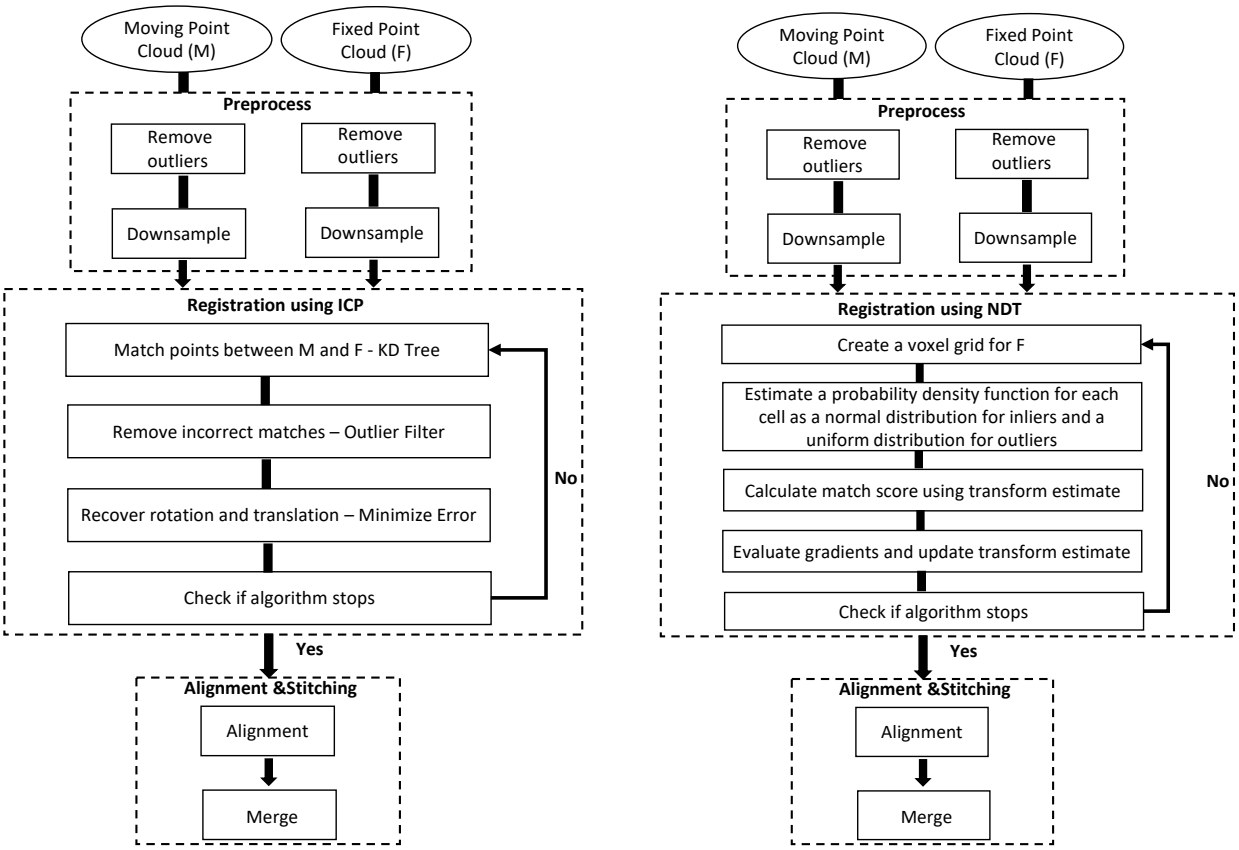


Figure 3.14. Workflows of pairwise point cloud registration based on (a) ICP and (b) NDT algorthms.

$$T_{i,i-1} = [R, t] \tag{3-13}$$

$$T_{j,1} = \prod_{i=2}^{i=j} T_{i,i-1} \tag{3-14}$$

3.6.2.2 Simultaneous groupwise registration

GlobalICP (Glira, 2015b) is used to optimize the alignment of multiple point clouds with the ICP algorithm, where a joint optimization of all the scans is implemented. A prerequisite is that all the point clouds should be approximately aligned, for which, sequential NDT is an optimal choice with high efficiency and acceptable registration accuracy. Consequently, the simultaneous groupwise registration scheme is indeed the combination of NDT and GlobalICP.

There are seven steps involved in the GlobalICP algorithm, which are shown in Figure 3.15 and specifically introduced by taking a pair of point clouds from observed data as an example (Figure 3.16). Firstly, the overlap area of the point clouds is determined by voxel hulls, then a subset of

points are selected within the overlap area in one of the point clouds, and the nearest neighbours of them in the other point cloud are found, forming a set of correspondences which might include several outliers. These outliers are removed later based on the compatibility of points. Afterwards, a weight between 0 and 1 is assigned to each correspondence according to the roughness attribute and the angle between normals of the corresponding points. The estimation of the transformation parameters (for the loose point cloud) is performed by a least squares adjustment which minimises the sum of squared point-to-plane distances. Finally, the loose point cloud is transformed with the estimated parameters. The above steps are iterated until the global minimum is reached.

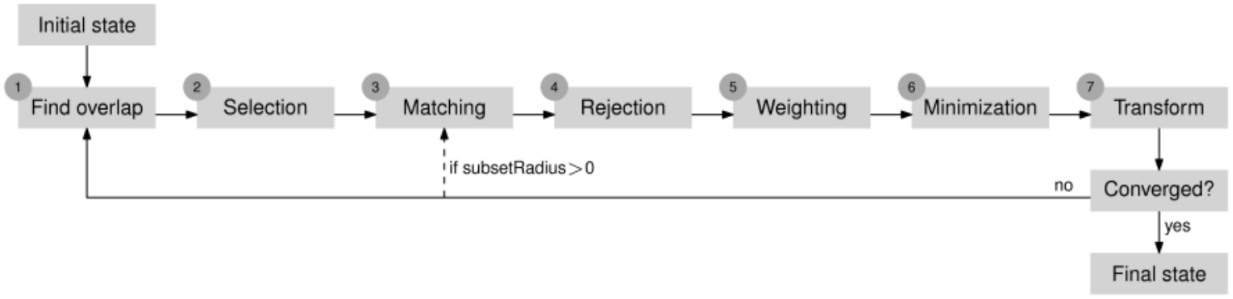
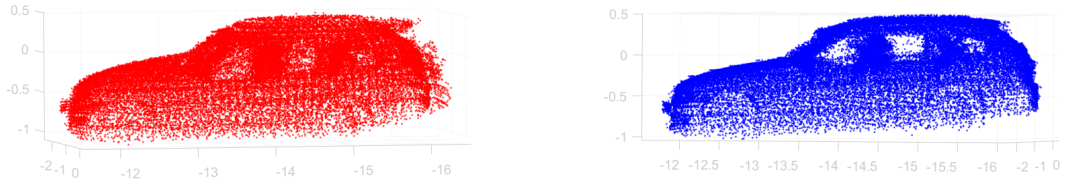


Figure 3.15. Functionality of the GlobalICP algorithm (Glira, 2015a)



(a) Initial state (b) Final state after using GlobalICP

Figure 3.16. The performance of GlobalICP

3.6.3 Measurement of vehicle dimensions

Vehicle size and vehicle type are two important parameters in vehicle emission studies according to Pinto et al. (2020). Vehicle reconstruction can benefit the acquisition of these two parameters. On one hand, vehicle dimensions can be measured from the reconstructed vehicle shape. On the other, it is easier to classify vehicles into different categories based on the complete vehicle shapes. Whilst the shape better identifies the vehicle type, key data such as Euro class, fuel type

and engine size that are needed to estimate emissions are not available. However, the improved classification better informs the National fleet, resulting in better estimates of emissions than those based on composition defined by loop detectors, for example.

3.6.3.1 Length and width

Figure 3.17 shows the plan-view of the reconstructed shapes of a vehicle example obtained from the four previously described methods. The shapes are rotated to a horizontal position to make it easier for the subsequent measurements. Vehicle length and width can be calculated from the minimum bounding box around the corresponding shape. It is noteworthy that the measurements from 2D matching method should be converted to real world scale through the pixelsize, as in Equation 3-7 in Section 3.4.2.

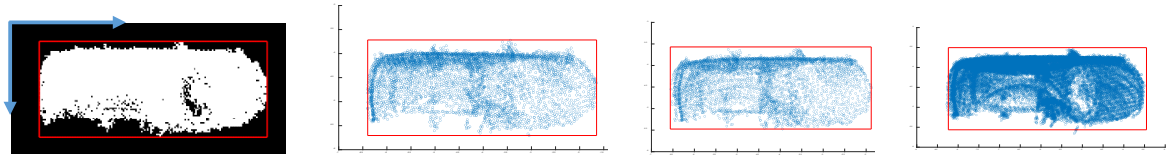


Figure 3.17. Measurements of vehicle shapes from different methods.

3.6.3.2 Height

The algorithm to calculate the vehicle height can be described by the following steps:

i) Label the road region manually and regress to a plane.

ii) Calculate the centroid of the vehicle and define a window around the centroid which is parallel to the road plane.

iii) Calculate the distance of each point within the window to the road plane.

iv) Find the maximum distance which is regarded as the vehicle height.

3.6.3.3 Construction of validation database

In order to validate four different vehicle reconstruction methods, a database containing 30 vehicles (two of them are shown in Figure 3.18) was created from video data simultaneously collected with lidar at the test sites. The make and model of each vehicle was identified according

to a voting strategy and its dimensions were acquired from manufacturer's specifications. Due to inadequate image resolution, it was not easy to clearly identify the make and model of the vehicles from the video imagery. To solve the problem, crowdsourcing strategy was adopted: 20 people at Newcastle University who are familiar with or greatly interested in vehicles contributed to the identification process. They named the make and model of each vehicle individually from their knowledge and experience. For each vehicle, there might be several makes, and models named. The final decision of it was the one with the highest voting number.

These vehicles were reconstructed by the introduced four methods from the lidar data. With cross-reference to the simultaneously acquired video imagery, the obtained vehicle dimensions were compared with those from the validation database.

Dimensions & Weight

Exterior

Length :	**3570**
Width :	**1630**
Height :	**1490**
Track, Front :	**1410 – 1420**
Track, Rear :	**1400 – 1410**
Wheel Base :	**2300 mm**
Turning Circle :	**9.3**
Ground Clearance :	**115 – 150**

(a) Fiat 500

Dimensions & Weight

Exterior

Length :	**3999**
Width :	**1713**
Width, with Mirrors :	**1944**
Height :	**1488**
Track, Front :	**1470**
Track, Rear :	**1460**
Wheel Base :	**2511 mm**
Overhang, Front :	**869**
Overhang, Rear :	**619**
Turning Circle :	**11**
Ground Clearance :	**125**

(b) Vauxhall Corsa

Figure 3.18. Two vehicle examples with identified dimensions.

3.6.4 Fine-grained vehicle classification

Fine-grained object classification has been increasingly studied for detailed 3D traffic scene understanding (Stark *et al.*, 2011; Lin *et al.*, 2014; Zia *et al.*, 2015). In terms of vehicles, it is important to acknowledge their types for vehicle emission studies, amongst other applications. Vehicles are first discriminated from other objects and then further classified into different categories. The two stage approach is usually reported to perform better than when vehicle detection and classification are performed as a general classification problem by also introducing a non-vehicle class (Serna and Marcotegui, 2014). Specifically, vehicles are intended to be classified into cars, vans, trucks and buses based on the observed data.

According to the proposed JDAT framework, moving objects are classified into vehicles and pedestrians by identifying the representatives of corresponding trajectories. For one trajectory, the representatives are further input to the vehicle reconstruction algorithm. The obtained vehicle shape is supposed to be sent to the classifier for fine-grained vehicle classification, for which the classifier should be trained by complete vehicle shapes reconstructed on trajectories obtained from the observed lidar data. Different from vehicle detection, RF is the only classifier adopted for vehicle classification since samples of each vehicle category are insufficient to train a multi-class PV-RCNN detector.

3.7 Summary

This chapter has introduced the proposed traffic monitoring system from the aspects of vehicle detection, tracking, and reconstruction. Both a three-step workflow and a deep learning method have been described in vehicle detection in Section 3.1 and Section 3.2. Traditional machine learning classifiers are applied in the final step of the first method and PV-RCNN is used as the vehicle detection network in the second method.

Fundamentals of tracking are firstly elaborated in Section 3.3, followed by two vehicle tracking frameworks (vehicle tracking and high accuracy speed estimation; joint vehicle detection and tracking). Vehicle detection in the first framework is realised by the aforementioned three-step vehicle detection workflow. Vehicle tracking is firstly conducted by a centroid-based initial

tracking procedure, and accuracy of the obtained vehicle speeds is further improved by a tracking refinement module. In the second framework, moving object clusters are extracted via the first two steps in the three-step vehicle detection workflow. Afterwards, object detection and tracking are performed in parallel, with PV-RCNN employed in detection and trajectory identity determined by the class of corresponding representatives. Four methods proposed from both 2D and 3D perspectives are used for vehicle reconstruction in Section 3.6. The applications of vehicle reconstruction including vehicle dimension measurement and fine-grained vehicle classification have been depicted at the end of Section 3.6. Experiments as well as comprehensive analysis and comparison of the above methods explained in the next Chapter 4.

Chapter 4. Experiments and Analysis

The lidar sensors and the study sites used in this research are introduced respectively in Sections 4.1 and 4.2, followed by the analysis of the experiments regarding each element of the proposed traffic monitoring system (Sections 4.3 to 4.6). In each section, datasets and important parameters are analysed in the first instance if there are any. Subsequently, the performance is evaluated and comprehensive comparisons are made to demonstrate advantages of the proposed methods. A thorough discussion is carried out to further assess the system and explore the potentials for improvement. The final section is a summary of this chapter.

4.1. Equipment

This research employs two lidar sensors. The first is a RS-LiDAR-32, a panoramic instrument from RoboSense. The sensor has a detection radius of up to 200m and is designed for various applications such as autonomous vehicles, robotics, and 3D mapping. It has 32 laser beams and collects data at a speed of 640,000pts/s. The scanning frequency is set to 10Hz in our tests. It covers a 360° horizontal FOV and a 40° vertical FOV with 15° upward and 25° downward looking angles. The second sensor is a Velodyne VLP-16, with 16 laser beams and a maximum detection range of 100m. The vertical field of view of the instrument is 30° with 15° upward and 15° downward. The scanning frequency is also 10Hz in our experiments. The reason why two sensors were used is as follows: Velodyne lidar is the most commonly used multi-beam lidar sensor around the world. However, Velodyne lidar sensors with the highest number of laser beams, such as HDL-32E and HDL-64E, were beyond the budget of this project, so a VLP-16 with 16 beams was chosen. From the initial experiments, 16 laser beams were not found to provide sufficient spatial detail on the vehicles for some required operations, such as vehicle reconstruction. Therefore, an alternative lidar sensor was needed. The Robosense RS-LiDAR-32, with 32 laser beams, provided a good choice when considering both the price and specifications.

The lidar viewer used to display the captured data for RS-LiDAR-32 is RSView, and that for VLP-16 is VeloView. CloudCompare is used to view a single lidar frame. A lidar frame can be displayed in different perspectives in CloudCompare, e.g. 3D view, plan view and side view. 3D view is

obtained when all the points are displayed with original X, Y and Z values. When Z value of each point is set to 0, plan view can be obtained. When X or Y value of each point is set to 0, the lidar frame is shown in side view. Attributes of two lidar sensors are detailed in Table 4.1.

Table 4.1. Attributes of two lidar sensors: VLP-16 and RS-LiDAR-32

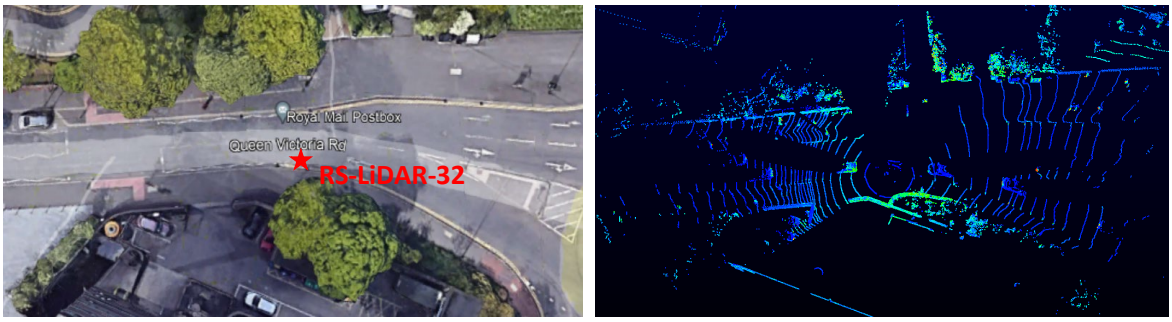
Attributes	VLP-16	RS-LiDAR-32
Horizontal FOV	360°	360°
Vertical FOV	30° (+15° to -15°)	40°(-25° to +15°)
Rotation Rate	5 - 20Hz	5 - 20Hz
Horizontal Angular Resolution	0.1°- 0.4°	0.09°- 0.36°
Vertical Angular Resolution	2°	At least 0.33°
Laser Emitters	16	32
Accuracy of Point Clouds	±3cm	±5cm
Measurement range	up to 100m	40cm to 200m

4.2 Study sites

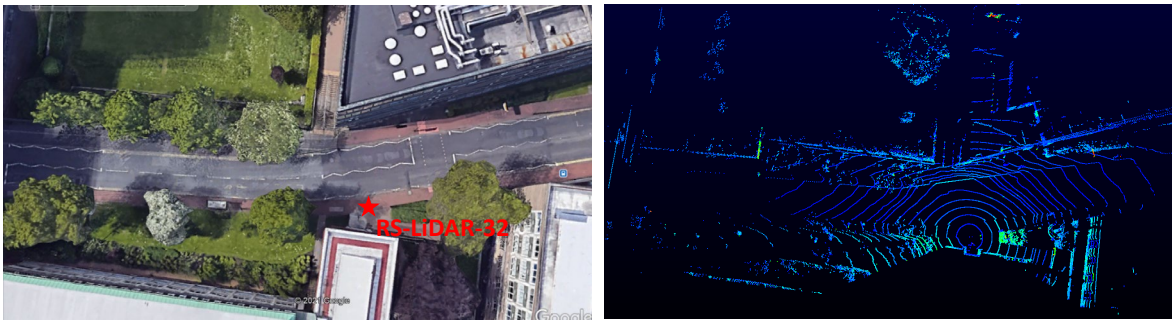
Four different study sites were chosen in Newcastle upon Tyne, UK (see Figure 4.1), to i) create datasets to train, validate or test vehicle detectors; ii) test methods or frameworks involved in the proposed traffic monitoring system under real-world traffic conditions. At Study Site 1, a RS-LiDAR-32 laser scanner was set up at a round corner along Queen Victoria Road, c.0.5m away from the first lane. At Study Site 2, a RS-LiDAR-32 laser scanner was set up along a straight road near a traffic light controlled pedestrian crossing. The lidar sensor was c. 2.5 m away from the first of two traffic lanes. At Study Site 3, a VLP-16 laser scanner was installed at a road intersection. It was c. 4.5m away from the first of multiple lanes. Study Site 4 was at a roundabout with busy traffic, where a VLP-16 was installed but with a shorter distance of c.2m to the nearest lane.

Table 4.2 displays the specific function of each study site. The reasons why different study sites have different functions are explained as follows:

(1) The initial vehicle detectors and the newly trained RF classifier were trained, validated and tested at Study Sites 1 to 3. Study Site 4 was not included because it provided similar vehicle data as Site 3.

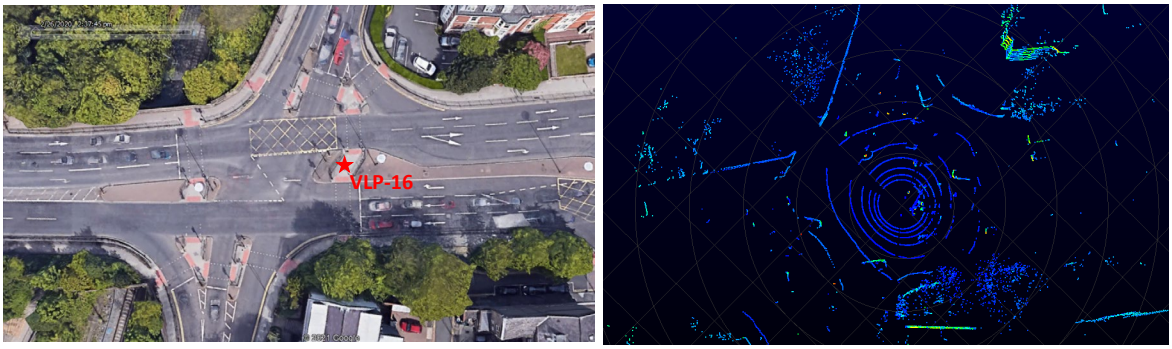


(a) Study Site 1: a section of Queen Victoria Road.



(b) Study Site 2: a single straight section of Claremont Road running through Newcastle University campus.

(c) Study Site 3: a junction of the Great North Road and St Mary's Place.



(d) Study Site 4: a crossroad of Clayton Road and Osborne Road in the region of Jesmond

Figure 4.1. Four study sites used in this research

Table 4.2. Functions of the study sites used in this research

Procedures		Study Sites	Functions
Vehicle detection	SVM, RF, rule-based	1,2,3	Model training, validation, test
	Newly trained RF	1,2,3	Model training, validation, test
	PV-RCNN	1,2	Model training, validation, test
Vehicle tracking	Framework 1	2,3	Algorithm test
	Framework 2	2,3,4	Algorithm test
Vehicle reconstruction		2	Algorithm test
Vehicle classification		1,2	Model training, validation, test

PV-RCNN was only performed on Study Sites 1 and 2. To train the network properly, a large training dataset was needed. Due to time limitation, it was difficult to create an adequate training dataset directly from Study Sites 3 and 4, or by adding a large number of samples from these two sites to the training dataset created from Sites 1 and 2. Moreover, the network trained using data from Study Sites 1 and 2 did not perform well when initially applied to Study Sites 3 and 4. Therefore, Study Sites 3 and 4 were not considered further with PV-RCNN at this stage.

(2) At Study Site 1, as the near lane was a bus lane, vehicles in the far lane were occluded whenever a bus passed by. Therefore, neither of the two tracking frameworks was tested at Study Site 1. Data from Study Sites 2 and 3 were sufficient to test framework 1. However, many more vehicle samples were required in order to test tracking framework 2, therefore Study Site 4 was used for this purpose.

(3) Vehicle reconstruction and fine-grained classification require sufficient vehicle details, so these algorithms were not assessed on Study Sites 3 and 4 where a lidar sensor of only 16 laser beams was installed.

Vehicle reconstruction was not tested on Study Site 1 because vehicles near the lidar sensor were distorted as the lidar sensor was located too close to the road edge. Also, as explained in (2), vehicles in the far lane were occluded whenever a bus passed by.

4.3. Vehicle detection

At the early stage of this research, classifiers such as SVM and RF were initially trained for vehicle and non-vehicle binary classification based on the small amount of data possessed at that time. Since the vehicle tracking and high accuracy speed estimation framework is developed based on vehicle detection results from these classifiers, they are introduced in this section, although are replaced by new classifiers at a later stage. RF, a three-class classifier aimed at categorising objects into vehicles, pedestrians and other classes, was trained with a larger dataset and more distinguishable feature sets when more data was available. An advanced 3D object detection network, PV-RCNN, was further exploited attempting to improve vehicle detection performance. In this section, the datasets used to train the classifiers are introduced in the first place, followed by the determination of important parameters involved in vehicle detection and illustration of the training process. Comprehensive comparisons regarding different classifiers or different operations of the same classifier are made to better show the advantages of the classifiers.

4.3.1 Datasets

For initial classifiers, a dataset containing 316 vehicle clusters and 224 non-vehicle clusters was created manually from lidar data collected at Study Sites 1 to 3. Around 67% of these were used for training the SVM and RF classifier (210 vehicles and 150 non-vehicles), and the remaining used for validation. A test dataset of 697 clusters (300 vehicles, 397 non-vehicles) that was totally new to the classifier was then randomly selected from the lidar observations at three study sites.

The classifiers in initial trials are for vehicle and non-vehicle binary classification. Since pedestrians (including a small number of cyclists and motorcyclists) are also important road users in addition to vehicles in urban cities, it is necessary to treat them as a single class with more traffic data available. In addition to road users, there are also some clusters from swaying trees or bushes in the extracted moving objects that need to be distinguished from vehicles and pedestrians. Therefore, a RF is trained for classification among three classes, i.e. vehicle, pedestrian and others. A dataset composed of 2928 clusters (1423 vehicle clusters, 1282 pedestrian clusters and 223 others) was created manually from lidar data collected at Study Sites

1 to 3 to train and validate the RF classifier. A five-fold cross validation strategy was used to split the dataset. A test dataset of 583 clusters (271 vehicle clusters, 306 pedestrian clusters and 25 others) that was totally new to the classifier was created from lidar observations at Study Site 2 and Study Site 3. It is noteworthy that both training and test datasets include data from two laser scanners in order to make the classifier more generalised.

For PV-RCNN, a dataset containing 3184 vehicles and 1563 pedestrians was created from 763 lidar frames collected at Study Site 1 and Study Site 2. 360 vehicles and 368 pedestrians from 63 frames (15 frames from Study Site 1 and 38 frames from Study Site 2) are composed of the test split. The remaining in the dataset are divided into train split and validation split by a ratio of 7:3. In our data, the numbers of cyclists and motorcyclists are so small that it is impossible to regard them as individual classes. Therefore, they are not considered in this study.

4.3.2 Parameters and training process

As for the three-step workflow, the background of each test site is constructed prior to moving point extraction (as described in Chapter 3). Background construction is normally conducted by successive frames in a certain time interval when the number of moving objects is as small as possible. In our experiments, for each of the four study sites, 100 successive frames in a quiet period were selected to perform background construction.

In clustering, there are three important parameters: the minimum cluster size S_1, the maximum cluster size S_2, and the minimum distance *d* between two clusters. At four study sites of this research, the minimum distance between two vehicles is around 1.5m, and the minimum distance between a pedestrian and a vehicle is around 1.8m. Therefore, d is set to 1m in the tests. The cluster size is dependent on the number of beams of the sensors, thus needs to be adjusted for different sensors. According to comprehensive statistics, the largest vehicle cluster contains around 6000 points from RS-LiDAR-32, so S_2= 6500. Since the point density is much lower from the VLP-16, the value is smaller: S_2= 5500. As for vehicle tracking and high accuracy speed estimation framework, the tracking refinement module requires clusters to be of certain shape and size. The smallest cluster that meets the requirements contains around 200 points at Study Site 1 and Study Site 2 where the RS-LiDAR-32 was installed, so S_1= 150 for these sites. A smaller

value S_1= 50 is adopted for Study Site 3 and Study Site 4 where the VLP-16 was installed. Regarding the joint vehicle detection and tracking framework, small clusters with few points in the far scanning field are supposed to be maintained because the following object tracking step is aimed to associate all the visible clusters in the scanning region so that tracking can be continued to the maximum extent. According to the datasets from four study sites, S_1 is set to 5. In the vehicle and non-vehicle classification stage, an SVM classifier with radial basis function as the kernel function and an RF classifier with 20 trees were exploited in the initial trials. Two classifiers were trained using Matlab 2020b on a laptop with i7-2.8GHz CPU.

As for PV-RCNN, the entire network was trained with batch size 4, learning rate 0.01 for 100 epochs on a NVIDIA GeForce RTX 3090 GPU, which took around 8 hours. The detection range is set to -80m~80m for the X axis, -40~40m for the Y axis and -3m~1m for the Z axis. Other parameters including voxel size in the training process remain the same for KITTI data as in the work of Shi *et al.* (2020a).

4.3.3 Evaluation Metrics

As pointed out by Zhang *et al.* (2020), three indices, precision, recall and F_1-score, were used to assess the performance of the classifiers. Macro F_1 is used to measure the overall performance of the newly trained RF regarding to 3 classes, vehicles, pedestrians and others, shown in Equation (4-1).

$$\text{Macro } F_1 = (F_{1\ (\text{vehicles})} + F_{1\ (\text{pedestrians})} + F_{1\ (\text{others})})/3 \quad (4\text{-}1)$$

To evaluate the performance of PV-RCNN, Average Precision (AP) (Padilla *et al.*, 2020) is also adopted in order to make comparisons with results from Shi *et al.* (2020a). According to Padilla *et al.* (2020), related concepts are explained as follows. N is the number of ground truths; IOU (Intersection Over Union) means the area of intersection over the area of union between the ground truth bounding box and the detected bounding box.

1) True Positive (TP): A correct detection with IOU $\geqslant$ threshold.

2) False Positive (FP): A wrong detection with IOU < threshold.

3) Precision: TP/(TP+FP).

4) Recall: TP/N.

5) F_1 score: 2* Precision*Recall/(Precision+Recall).

4.3.4 Results and analysis

The results of the initial trials are analysed firstly. A RF classifier for vehicle and non-vehicle classification using the first 28 features introduced in Section 3.1.3 was trained and the weights for each of the features were obtained, shown in Figure 4.2. It can be concluded that a, $Width$, x_s, $Length$ and y_s are the five features with high importance, among which a is the most important one. As a result, we retrained the SVM and RF classifiers using the following feature sets: $F_1 = a$, $F_3 = [a, Width, x_s]$, $F_5 = [a, Width, x_s, Length, y_s]$, $F_{28} = [v_1, v_2, \dots, v_{20}, x_s, y_s, z_s, Length, Width, Max\_height, Min\_height, a]$. In addition, a simple rule-based method using the size of the clusters' bounding boxes was implemented for comparison with these two classifiers. Three indices, precision, recall and F_1-score, were used to assess the performance. From comparison results shown in Table 4.3, it is found that the overall performance of the three classifiers on four feature sets can be regarded as relatively indistinguishable. Both SVM and RF performed slightly differently with different number of features. When the feature set changes from F_1 to F_{28}, F_1 scores produced by SVM are 0.91, 0.92, 0.88 and 0.95, respectively. F_1 scores from RF are 0.91 for feature set F_1, and 0.93 for feature sets F_3, F_5 and F_{28}. The rule-based method performed best in terms of recall (0.99) with a decent precision as 0.92. In order to keep as many vehicles as possible to facilitate the tracking process, the rule-based method is adopted in the vehicle tracking and high accuracy speed estimation framework even though its overall performance is not the best.

As for results from the newly trained RF, it can be seen from Table 4.4 that for vehicle and pedestrian, F_1 is over 0.9 for both validation and test datasets. However, macro F_1 is lower than 0.9 due to low F_1 values of 'others' (0.83 for validation dataset and 0.75 for test dataset). To be specific, the precision of 'others' is comparable with the other two classes (0.88 and 1 for validation and test, respectively), whereas recall is much lower (0.79 for validation and 0.6 for test). The poor performance results from the limited number of clusters belonging to others (mainly refer to false alarms) in the training dataset. Since the number of false alarms in the test

datasets from Study Sites 2 and 3 is quite small, this classifier is regarded as competent to distinguish vehicles and pedestrians.

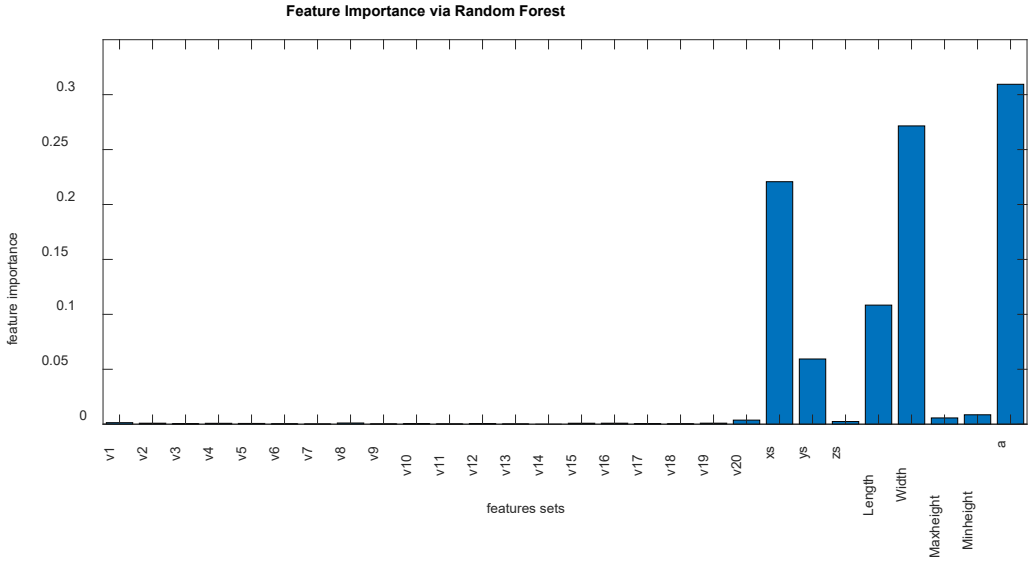


Figure 4.2. Estimation of feature importance.

Table 4.3. Peformance of classifiers trained by different feature sets.

	SVM			RF			Rule-based		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
F_1	0.86	0.96	0.91	0.90	0.92	0.91	0.86	0.99	0.92
F_3	0.88	0.96	0.92	0.91	0.95	0.93			
F_5	0.78	1.00	0.88	0.92	0.94	0.93			
F_{28}	0.96	0.94	0.95	0.90	0.96	0.93			

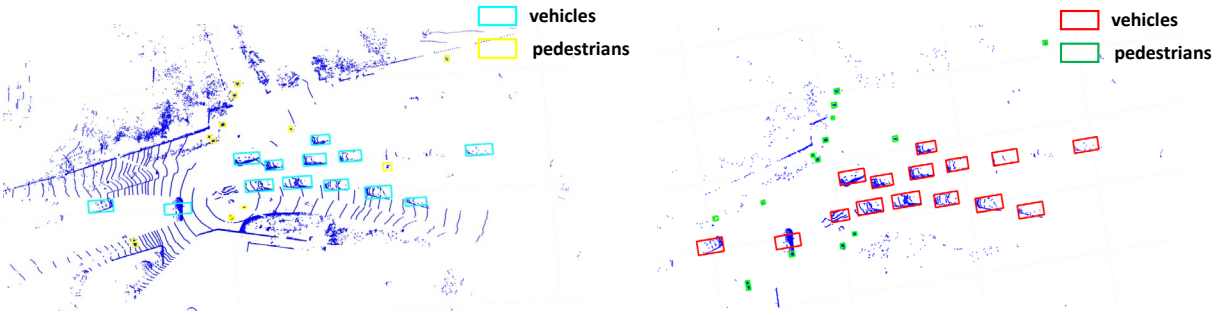
Table 4.4. Performance of RF classifier.

	Classes	P	R	F_1	Macro F_1
Validation	Vehicles	0.92	0.93	0.92	0.89
	Pedestrians	0.92	0.92	0.92	
	Others	0.88	0.79	0.83	
Test	Vehicles	0.86	0.95	0.90	0.85
	Pedestrians	0.95	0.87	0.91	
	Others	1.00	0.60	0.75	

The performance of PV-RCNN is analysed as below:

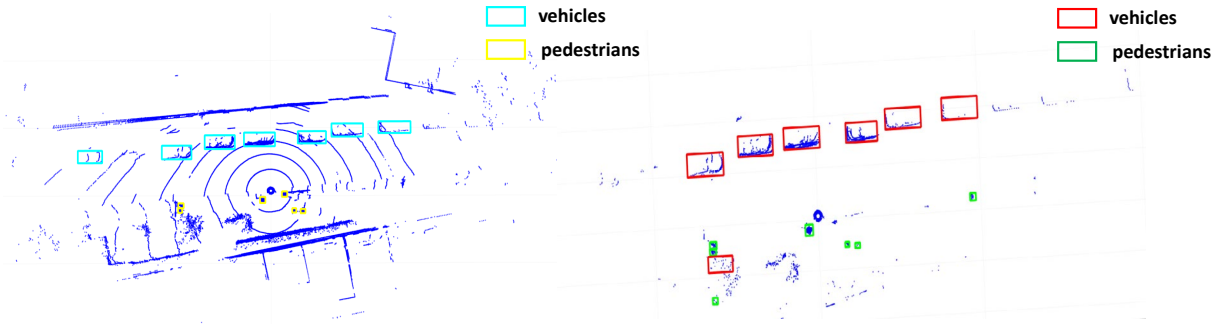
In the KITTI benchmark, cars, pedestrians and cyclists are detected. However, in our study, cyclists in the observed data are so few in number that it is impossible to regard them as an individual class. They are regarded as pedestrians in the current work because they are closer to pedestrians than to vehicles in appearance. In addition to cars, there are other categories in the observed data such as bus, truck and van. These categories are all regarded as vehicles currently due to data limitation. Unlike the newly trained RF, the class 'others' is not taken into consideration for PV-RCNN due to insufficient samples. Only vehicles and pedestrians are detected by PV-RCNN in this study.

As can be seen from Table 4.1, PV-RCNN has been tested at Study Site 1 and Study Site 2. The results are shown in Figure 4.3 and Figure 4.4. Table 4.5 displays the corresponding statistics. For vehicle class, the results outperform the reported accuracy from Shi *et al.* (2020a) with obvious margins, *i.e.* AP for vehicle at Site 1 is 96.6%, higher than the reported 90.3% for easy cars. At Site 2, AP for vehicle is 85.0%, higher than the reported accuracy 81.4% for moderate cars (Shi *et al*. 2020a). For pedestrian class, AP values at two sites are 78.7% and 54.2%, higher than the reported accuracy 52.2% for pedestrians (Shi *et al*. 2020a). What needs to be addressed is that vehicle class in our data includes various categories such as car, van, bus and truck, which makes vehicle and pedestrian classification more difficult than single category. Despite this fact, PV-RCNN achieves better results for vehicle class on our data than car class on KITTI data. F_1 of vehicle at Site 1 is 88.8%, 10.6% higher than that at Site 2. F_1 of pedestrian at Site 1 is 6.5% higher than that at Site 2. At both Sites, F_1 of pedestrian is more than 10% lower than that of vehicle. The poorer performance at Study Site 2 is resulted from the smaller amount of training samples. The worse results for pedestrians can be explained as proposed by Shi *et al.* (2020a): the limited number of key-points may harm the performance of objects with small sizes.



(a) Ground truths (b) Detection results

Figure 4.3. PV-RCNN detection results of Study Site 1.



(a) Ground truths (b) Detection results

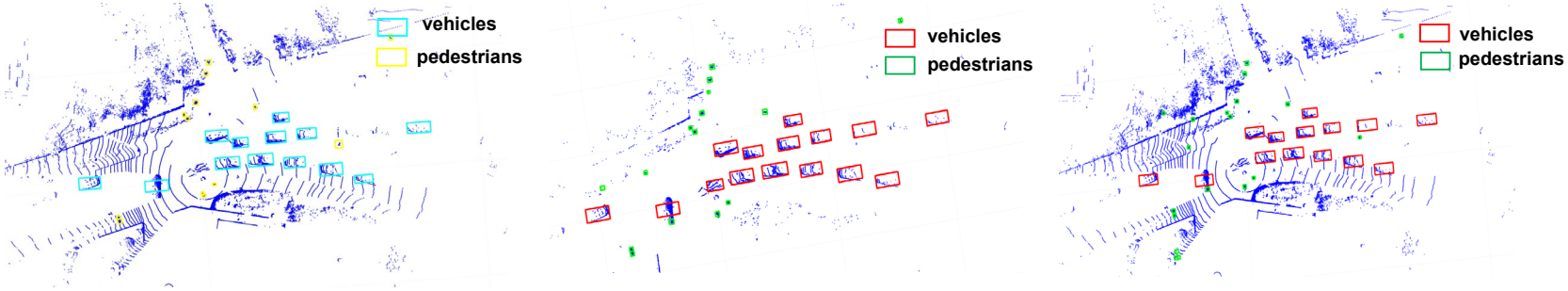
Figure 4.4. PV-RCNN detection results of Study Site 2.

Table 4.5. Statistics of detection results from PV-RCNN.

Study Sites	Class	N	TP	FP	F_1 (%)	AP(%)	IoU
1	Vehicle	178	174	40	88.8	96.6	0.5
	Pedestrian	222	200	121	73.7	78.7	0.5
2	Vehicle	182	158	64	78.2	85.0	0.5
	Pedestrian	146	89	30	67.2	54.2	0.5

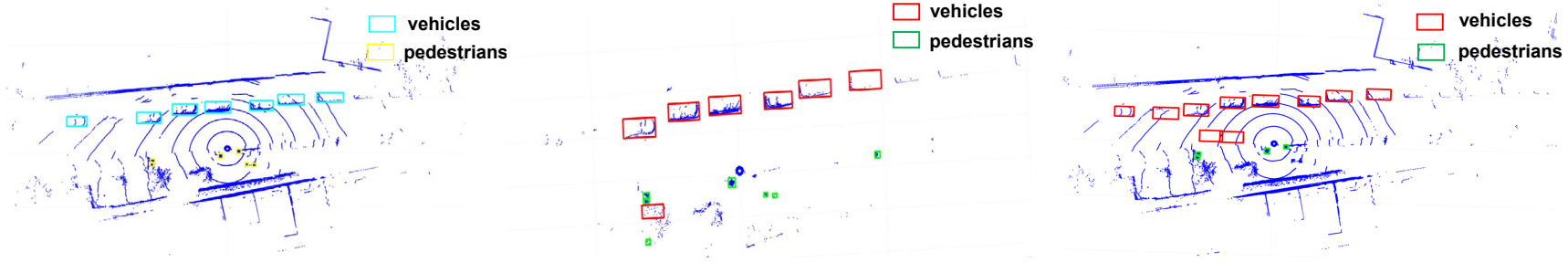
4.3.5 Comparison between PV-RCNN on original lidar data and moving points

PV-RCNN was developed on original lidar data as an object detector by Shi *et al.* (2020a). However, it was operated on extracted moving points in this research to avoid some potential false alarms by removing unrelated points beforehand. Figure 4.5 and Figure 4.6 show the comparison of the results, and Table 4.6 displays corresponding statistics.



(a) Ground truth (b) Detection on moving points (c) Detection on original data

Figure 4.5. Comparison results of Study Site 1.



(a) Ground truth (b) Detection on moving points (c) Detection on original data

Figure 4.6. Comparison results of Study Site 2.

Table 4.6. Comparison between PV-RCNN on original data and moving points.

Sites	Data	Class	N	TP	FP	Recall (%)	Precision (%)	F_1 (%)
1	Original	Vehicle	178	174	40	97.8	81.3	88.8
		Pedestrian	222	200	121	90.1	62.3	73.7
	Moving points	Vehicle	178	177	41	99.4	81.2	89.4
		Pedestrian	222	192	69	86.5	73.6	79.5
2	Original	Vehicle	182	158	64	86.8	71.2	78.2
		Pedestrian	146	89	30	61.0	74.8	67.2
	Moving points	Vehicle	182	136	27	74.7	83.4	78.8
		Pedestrian	146	106	56	72.6	65.4	68.8

It can be seen from Table 4.6 that at two study sites, performance of vehicle and pedestrian classification has been improved by performing PV-RCNN on moving points. To be specific, at Study Site 1, F_1 of vehicle has increased from 88.8% to 89.4%, while F_1 of pedestrian has reached 79.5% from 73.7%. At Study Site 2, F_1 of vehicle is slightly enlarged by 0.6%. The increment of F_1 for pedestrian is 1.7%. It is noteworthy that the improvements of pedestrian (5.8% and 1.7%) are more obvious than those of vehicle (0.6% and 0.6%). The main reason is that unrelated points in the original data has a bigger negative impact on small size objects such as pedestrians, thus the removal of those points benefits more for pedestrians than for vehicles. The increments of F_1 for vehicle and pedestrian are smaller at Study Site 2 than at Study Site 1. As can be seen from Figure 4.5 and Figure 4.6, the environment of Study Site 1 is more complicated than that of Study Site 2, therefore, simplifying the data by removing static background makes bigger contributions to object classification at Study Site 1 than at Study Site 2. The comparison and analysis demonstrate the advantages of performing PV-RCNN on moving points over original lidar data in terms of vehicle and pedestrian detection.

4.4 Vehicle tracking and high accuracy speed estimation

4.4.1 Parameter analysis

Some important parameters in the centroid-based tracking stage of the vehicle tracking and high accuracy speed estimation framework are specified in Table 4.7. These parameters are involved in three stages of tracking including initialization, data association and track management. The description, setting as well as the reason of the setting for each parameter is shown in the table. 'Initialization threshold' is used to start a new track. If the association probability of a detection within the assignment gate is lower than the threshold, a new track will be generated. This parameter is usually set as a scalar in [0,1]. In this study, the default value 0.1 in JPDAF algorithm was assigned to this parameter. 'Confirmation threshold' is a parameter to confirm a track and normally specified as [M, N]. A track is confirmed if it recorded at least M hits in the last N updates. Thus, the first M-1 clusters of an object are not assigned to the corresponding track. To avoid missing any potential targets, the confirmation threshold in this study was set to [1,3]. 'Assignment threshold' is the pivotal parameter in data association. It controls the range in which the detections are assigned to tracks, namely, the assignment gate. If the value is too small, some detections that should be assigned to a track might be overlooked. Otherwise, there will be false assignments. In this study, it was set to 4m, considering both the average vehicle speed and lidar sensor frame rate.

There are two parameters in track management worth mentioning: the first one is 'Deletion threshold', used to delete a track. It is normally set as [P,R], which means a confirmed track will be deleted if it is not assigned to any detection in P of the last R tracker updates. The default value in the JPDAF algorithm is [5,5] and this value was adopted in this study. The other one is 'Length threshold', a parameter used to delete trajectories that do not belong to road users. It was set to 3m with the following considerations: according to the data, non-road users are mainly moving tree leaves that could not be removed by moving point extraction. A bunch of such tree leaves compose a big cluster. There are several such clusters in the processed data. Based on experimental tests, the maximum diameter of those clusters is around 3m. In practice, the trajectories of moving on-road users must be longer than 3m and the observation lasts for a

certain period (The minimum recording time in this research is 42s, and no users remain static throughout the recording).

Table 4.7. Parameter setting in the centroid-based tracking stage.

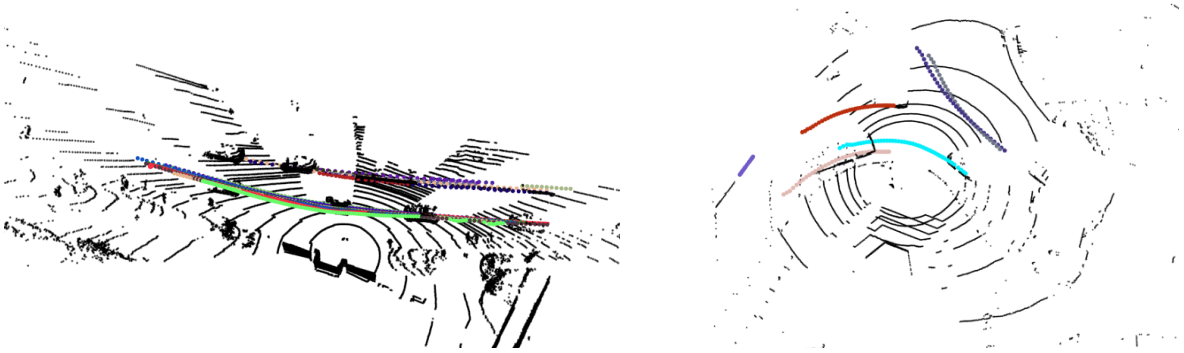
Procedure	Parameter	Description	Setting	Basis of setting
Initialization	Initialization threshold	Threshold to initialize a track	0.1	Default
	confirmation threshold	Threshold for track confirmation	[1,3]	Experiment
Data association	Assignment threshold	Detection assignment threshold	[4, Inf]	Practice and empirical knowledge
Track management	Deletion threshold	Threshold for track deletion	[5,5]	Default
	Length threshold	Threshold to delete a non-vehicle trajectory	3	Experiment and practice

4.4.2 Vehicle tracking performance

In order to validate the speed estimation under different traffic flow conditions at Study Sites 2 and 3, six recordings covering the test vehicle when it was passing through the scanning area—three from each site—were taken as three study cases. As described in the experimental design of the speed reference system (Section 3.4.3), a recording is defined as: when the test vehicle is near the scanning area, recording starts. After the test vehicle leaves the scanning area, recording stops. The six study cases were not selected at random. The first three were from Study Site 2. Date collection was conducted in the afternoon when the traffic was busy. Even though six recordings in total were collected at Study Site 2, not all of them can be properly used because of heavy occlusion. Three were used with the following reason: the test vehicle in case 1 and case 2 showed different dynamics, with one driving through and the other showing a pattern of stop-and-go. The lidar sensor was set to a different angle of view in case 3, as an aid to understand how best to configure the optimal vertical FOV. The other three were from Study Site 3. Date collection was conducted at noon when the traffic was freely flowing. Six recordings were

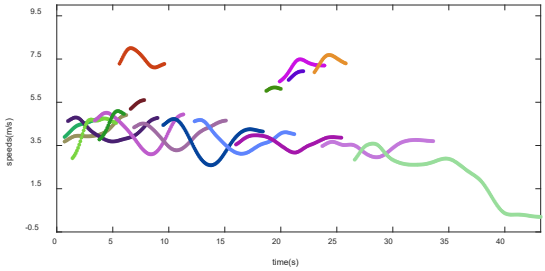
available to be used from this site, but only three representatives were used in order to reflect different patterns of movement: turning right, turning left, driving straight forward. In each of the six cases, two sets of vehicle speeds were acquired through the tracking process: initial values from centroid-based tracking and refined values after tracking refinement. Both were compared with the reference datasets so as to assess the accuracy improvement. Case 1 and case 5 are illustrated in detail.

Figure 4.7 and Figure 4.8 show the trajectories and speeds of all the tracked vehicles in case 1 and case 5. In case 1, the recording was shown for 43.3s and tracking continued through the whole period, with all 18 vehicles that appeared in the scanning area during this period successfully tracked. The tracking range of the approach is c. 45m. Short trajectories were generated from vehicles that were either close to the edge of the scanning area at the beginning of the observation period, or that were occluded by other vehicles during tracking process. In case 5, tracking was shown c. 2s before the test vehicle entered the scanning range and stopped c. 2s after it left. The whole process lasted for c. 8s and all six vehicles (including one bus) were tracked. Two vehicles that were turning left were continuously tracked. Tracking of another two vehicles which were turning right, including the test vehicle, were also successively implemented. The trajectories of the remaining vehicles were very short, for example, one in the second lane. The successful tracking range in this case was c. 18m.

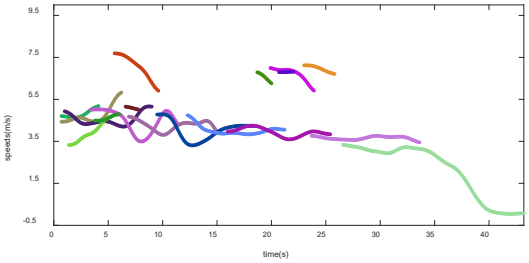


(a) (b)

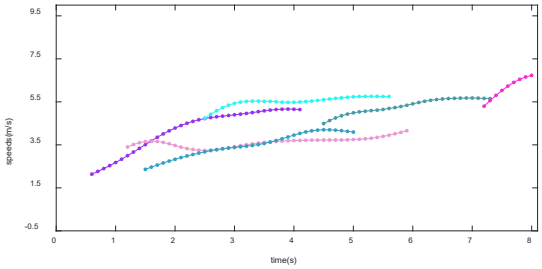
Figure 4.7. Trajectories of centroid-based tracked vehicles in case 1 (a) and case 5 (b). Each colour represents a vehicle with a unique ID.



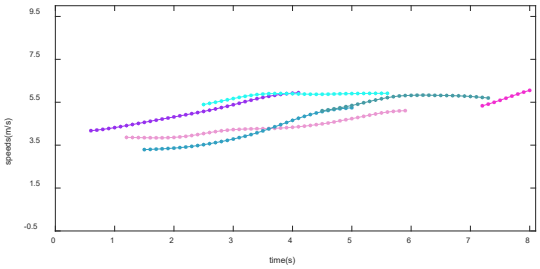
(a) Speeds of all tracked vehicles in case 1 (Centroid-based)



(b) Speeds of all tracked vehicles in case 1 (Refined)



(c) Speeds of all tracked vehicles in case 5 (Centroid-based)



(d) Speeds of all tracked vehicles in case 5 (Refined)

Figure 4.8. Speeds of tracked vehicles in case 1 and case 5: (a) and (c) show the centroid-based speeds; (b) and (d) show the refined speeds. Each colour represents a vehicle with a unique ID.

4.4.3 Evaluation of the reference speeds

The uncertainty of vehicle speeds from the reference system was evaluated, which was first implemented when the test vehicle was stationary so that the true speed was known to be 0 m/s. Two sections (shown in Figure 4.9) from Study Site 2 were chosen to create the statistics based on the following indices: Standard Deviation (SD), Mean, and Root Mean Square Error (RMSE). In the two cases, the durations when the vehicle was stationary were 23s and 41s, respectively. As shown in Table 4.8, the average RMSE of two cases was 0.024m/s, which means the displacement deviation is within 3mm per frame (frame rate 0.1s).

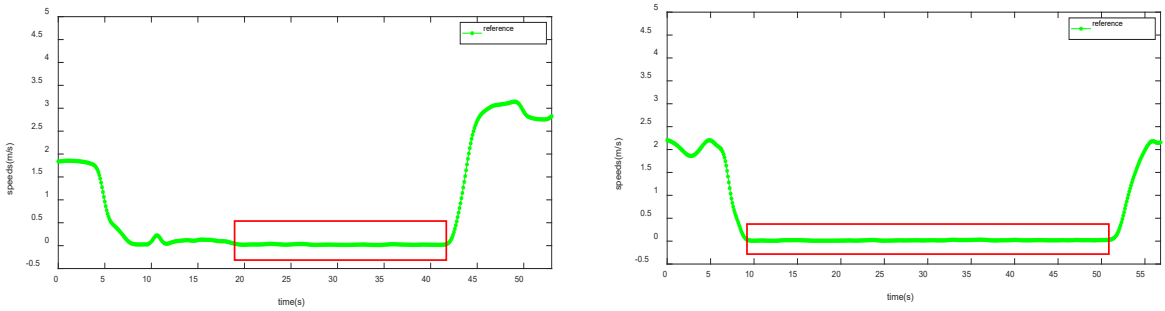


Figure 4.9. Reference speeds from two stationary sections.

Table 4.8. Analysis of reference speeds in two stationary sections.

	SD(m/s)	Mean(m/s)	RMSE(m/s)	Stationary period(s)
Section 1	0.006	0.023	0.028	19 to 42
Section 2	0.005	0.019	0.020	10 to 51

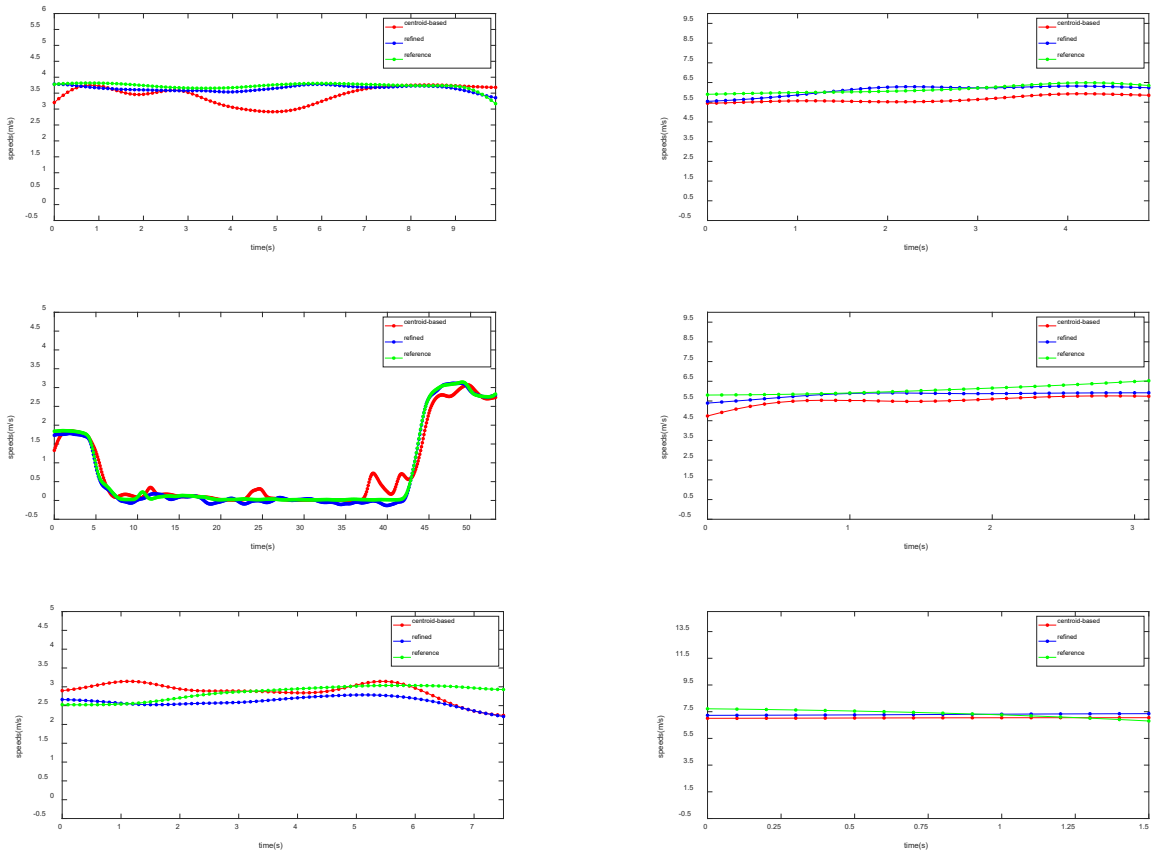
4.4.4 Comparison among three sets of vehicle speeds

Comparison among three sets of speeds (the centroid-based speeds; the refined speeds; the reference) was conducted for all the six cases. Figure 4.10 shows the comparison results of the test vehicle in six cases at Study Site 2 and Study Site 3. The test vehicle at Study Site 2 was either moving forward with a constant speed or with a pattern of "stop-and-go." The test vehicle at Study Site 3 was turning left and right in the first two cases, while going forward with a constant speed in the third case. It is also noteworthy that the test vehicle was going generally faster at Study Site 3 than at Study Site 2. Despite the movement variability, the refined speeds and the reference are in close accordance with the roughly estimated moving trends, whereas centroid-based speeds obviously deviate further from the reference. Therefore, it is clear that the tracking refinement step has improved the speed accuracy for all six example cases.

RMSE and mean absolute error (MAE) are used to quantitatively evaluate the three sets of speeds. As seen from Table 4.9, the accuracy of the estimated speeds was improved by the refinement module, under different vehicle dynamics. The overall mean RMSE value has decreased from 0.4m/s to 0.2m/s and the overall mean MAE has decreased from 0.3m/s to 0.2m/s. Meanwhile, the two different lidar sensors employed at the two study sites produce slightly different results, with RMSE value of 0.2m/s at Study Site 2 and 0.3m/s at Study Site 3.

However, direct comparison of performance between study sites is difficult since there are mainly four primary variables relating to the experiment configuration that might influence the achieved RMSE at different sites. These are: object distance to lidar sensor, number of lidar laser beams, vehicle speed and vehicle movement patterns. The test vehicle was observed at different distances to the lidar sensor at the two study sites: the test vehicle was approximately 10m to 15m from the sensor at Study Site 3; while the vehicle was only around 5m from the sensor at

Study Site 2. Also, the lidar sensor used at Study Site 3 had only 16 laser beams and that used at Study Site 2 had 32 beams. A further distance to the lidar sensor and fewer laser beams may result in deficiency in acquiring points on the vehicles, particularly the roofs. Speed estimates with lower precision and accuracy may therefore be obtained as there is insufficient detail in the image matching process during tracking refinement.



(a) (b)

Figure 4.10. Comparison results of the test vehicle in six cases at Study Site 2 and Study Site 3: red: centroid-based speeds; blue: refined speed; green: the reference. (a) Test vehicle speeds in case 1 to case 3. (b) Test vehicle speeds in case 4 to case 6.

Other factors that may influence RMSE values are vehicle speeds and movement patterns. The faster the vehicle is being driven, the smaller the overlap area between two successive vehicle clusters will be. As long as the overlap area accounts for a sufficient percentage of the whole vehicle body such that that matching can be performed smoothly, the accuracy of the estimated vehicle speeds would not be adversely affected. It is worth exploring the relationship between RMSE and the required percentage of overlap. In theory, the moving pattern would not influence

the matching performance as long as there is no sudden motion change. When the vehicle is turning right or left smoothly, matching would still be achieved within high quality. Unfortunately, due to time constraints, data collection was curtailed. A small number of data samples has limited statistical rigour and therefore more robust analysis would require further data collection.

Table 4.9. Evaluations of six case studies.

		RMSE(m/s)		MAE(m/s)		Mean speed of test vehicle (m/s)	Vehicle travel direction
		$RMSE_{13}$	$RMSE_{23}$	MAE_{13}	MAE_{23}		
Study Site 2	Case 1	0.4	0.1	0.3	0.1	3.7	Straight on
	Case 2	0.2	0.1	0.1	0.1	0.7	Straight on
	Case 3	0.4	0. 3	0.3	0.3	2.6	Straight on
Mean		0.3	0.2	0.2	0.1		
Study Site 3	Case 4	0.5	0.2	0.5	0.2	6.1	Turning left
	Case 5	0.5	0.3	0.4	0.2	5.8	Turning right
	Case 6	0.4	0.3	0.4	0.3	7.3	Straight on
Mean		0.5	0.3	0.4	0.2		
Overall mean		0.4	0.2	0.3	0.2		

4.4.5 Comparison with other methods

The estimated speeds were validated against a reference system (seen as Figure 3.10 in Section 3.4.3) that is considered to provide a higher order of accuracy. The RMSE of the reference data was about one-tenth of that of the lidar data. In the work from Zhao *et al.* (2019), speed validation was conducted by a test vehicle with an on-board diagnostics logger. The average absolute speed difference between speeds from lidar data and reference data, which is equivalent to MAE in the study of this thesis, is as high as 0.6m/s. In comparison, the average MAE of all the cases in the work reported here was 0.2m/s. A more accurate reference system allowed full exploration of the capacity of lidar speed estimation.

In a vision-based vehicle tracking method (Bell *et al.*, 2020), four cases were from Study Site 2 in the study of this thesis and the adopted speed validation system was the same as that in this study. The reported average RMSE and average MAE values of four cases were 0.6m/s and 0.5m/s, around three times of corresponding values in Table 4.9 (0.2m/s and 0.2m/s). Therefore, it can be concluded that the proposed lidar-based vehicle tracking and speed estimation method performances better than vison-based method.

4.5 Performance of JDAT

The most important parameters in this framework are determined in Section 4.5.1. A segmentation-tracking-classification (STC) scheme is implemented to make comparison with the proposed framework in Section 4.5.2, in order to demonstrate that the proposed framework has stronger ability of trajectory classification. STC is a framework to track and identify vehicles with the same intention, as of the proposed JDAT, to mitigate negative influence from vehicle detection in traditional tracking-by-detection method. A tracking-by-detection method is conducted in Section 4.5.3 to show that the proposed framework has the ability to improve the quality of object trajectories

4.5.1 Parameter analysis

Settings for critical parameters in object detection based on PV-RCNN have been illustrated in Section 4.3.2. Parameter settings in object tracking can be seen in Table 4.7 in Section 4.4.1. There are two important parameters in trajectory classification: n, the number of representatives of a trajectory; p, the ratio of those identified as vehicles to all the representatives. Six study cases, the first three from Study Site 2, the fourth and fifth from Study Site 3 and the last from Study Site 4, have been tested with different values (n= [10, 20, 30], p= [0.5, 0.6, 0.7, 0.8]) to decide the optimal n and p with regard to the trajectory classification performance which is measured by the F_1 score. According to Figure 4.11, the best classification performance can be achieved when n=30 and p=0.5.

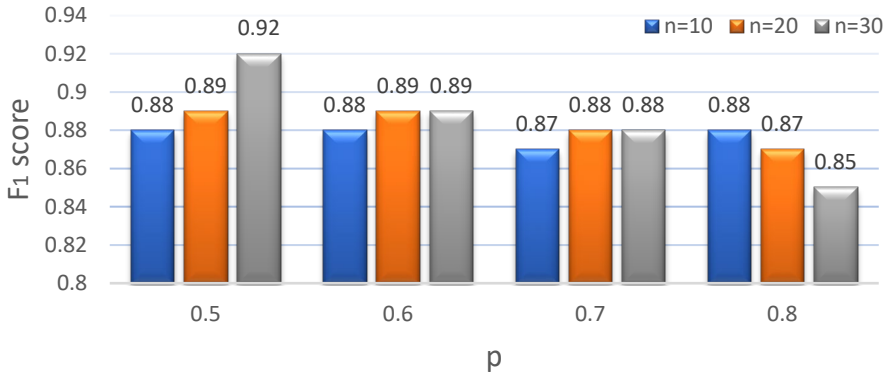


Figure 4.11. Classification performance with different n and p values.

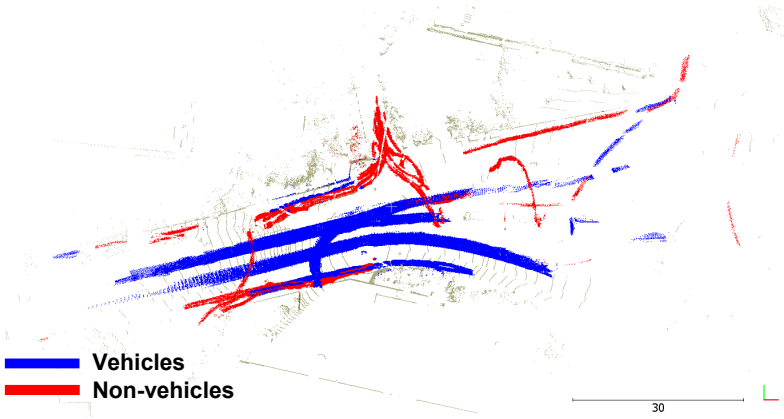
4.5.2 Comparison with STC scheme

In STC scheme, three procedures including moving object segmentation, object tracking and trajectory classification (RF is utilized) are conducted in sequence. n=30, p=0.5 are used in trajectory classification stage in both methods.

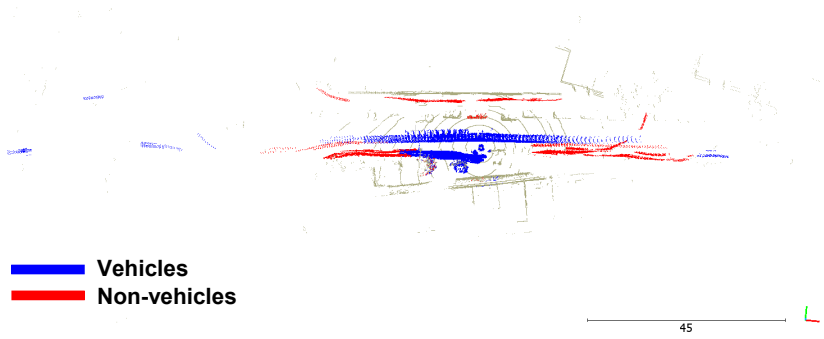
Trajectory classification results from two methods for two study sites are shown in Figure 4.12. The corresponding statistics are shown in Table 4.10. The indexes TP, FP and F_1 are defined in 4.3.3. In this situation, TP means the number of correctly classified trajectories; FP refers to the number of falsely classified trajectories; Ground Truth (GT) is the total number of trajectories of each class, which is manually identified from the tracking results. Although RF used in STC is trained to classify objects into vehicles, pedestrians and others, the last two classes are combined into one class, 'non-vehicle', in the statistics since the number of 'others' is pretty small at Study Site 1 and Study Site 2. Overall, the proposed method outperforms STC in trajectory classification regarding both vehicles and non-vehicles. Specifically, at Study Site 1, F_1 of vehicle from the proposed method is 83.6%, 3.6% higher than that from STC. F_1 of non-vehicle is increased by 2% by the proposed method. At Study Site 2, F_1 for both vehicle and non-vehicle have been increased by larger percentages than Study Site 1 which is located in a more complicated environment, 9% for vehicle and 14% for non-vehicle. The results demonstrate that the proposed method outperforms STC in representative-based trajectory classification especially in simpler environments.

Table 4.9. Comparison between STC and JDAT

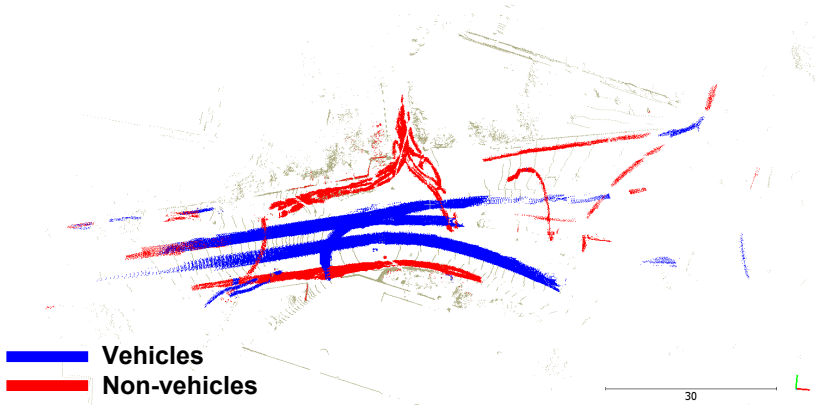
	Study Sites	STC			JDAT			GT
		TP	FP	F_1 (%)	TP	FP	F_1 (%)	
Vehicle	1	26	8	80.0	28	8	83.6	31
	2	18	16	66.7	14	3	75.7	20
Non-vehicle	1	36	5	84.7	36	3	86.7	44
	2	29	2	76.3	42	6	90.3	45



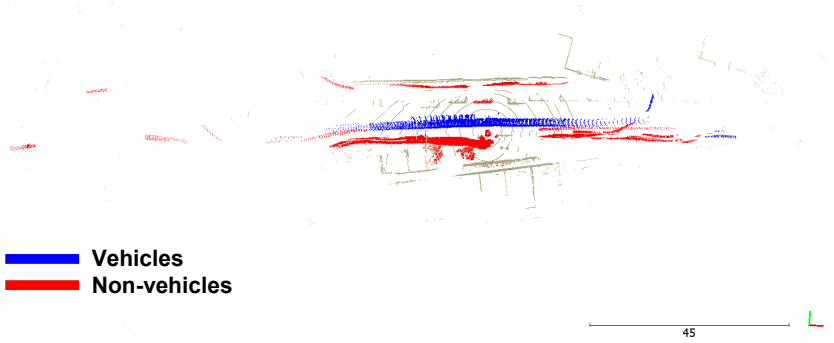
(a) trajectory classification from STC at Study Site 1



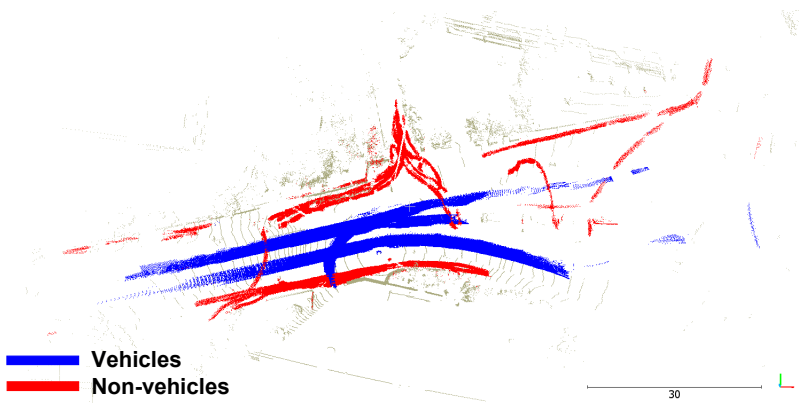
(d) trajectory classification from STC at Study Site 2



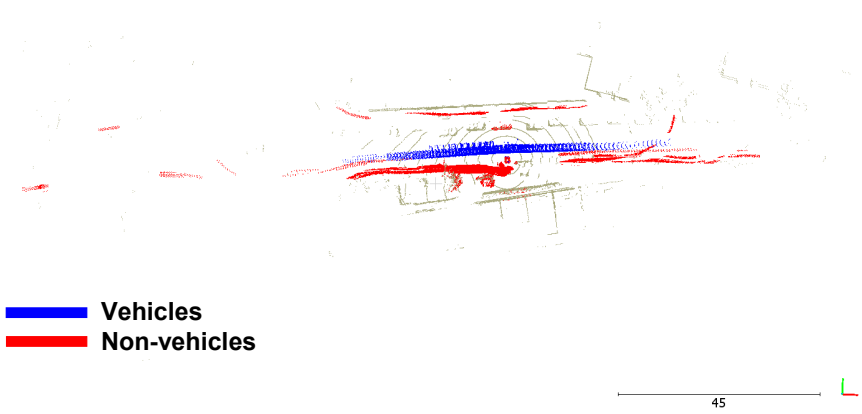
(b) trajectory classification from JDAT at Study Site 1



(e) trajectory classification from JDAT at Study Site 2



(c) Ground Truth at Study Site 1



(f) Ground Truth at Study Site 2

Figure 4.12. Trajectory classification performance of STC and JDAT.

4.5.3. Comparison with tracking-by-detection method

In the tracking-by-detection method, tracking is implemented after the objects are detected. A total of 15 vehicle examples from Study Sites 2, 3 and 4 were used to compare the proposed method with the tracking-by-detection method with regard to both the range and the continuity of the trajectories. Nine vehicle examples, three from each site, were used to compare the ranges of the trajectories. The other six vehicle examples, two from each site, were used to compare continuity of the trajectories. One vehicle example refers to the case in which a vehicle enters the scanning region, then passes by the lidar sensor, and finally leaves the scanning region. Different patterns of movement, i.e. turning right, turning left and driving straight forward, were involved in the 15 vehicle examples. The maximum tracking ranges of two commonly used lidar sensors are further measured. The trajectories of these vehicle examples are shown in Figure 4.13 to Figure 4.19, and the statistics about ranges of the first nine examples are displayed in Table 4.11.

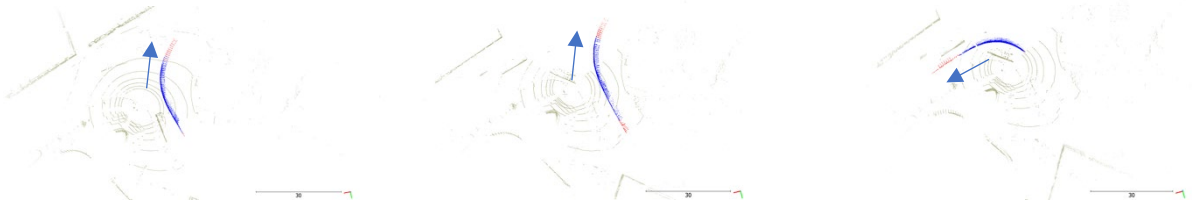
4.5.3.1 Ranges of the trajectories

Nine vehicles travelling across the entire scanning region are used to compare two methods. For each vehicle, two trajectories are obtained from two methods, individually. The start frame and the end frame of each trajectory are recorded in Table 4.11, so is the total number of frames the trajectory covers which is denoted as N_1 for the tracking-by-detection method and N_2 for the proposed method. By checking the vehicle clusters from the original data, the corresponding ground truth (the number of frames is denoted as N in Table 4.11) which refers to the frames where the vehicle actually exists can be obtained and regarded as the reference to compare the performance of two methods. N_1/N, N_2/N are used as indexes for the comparison.

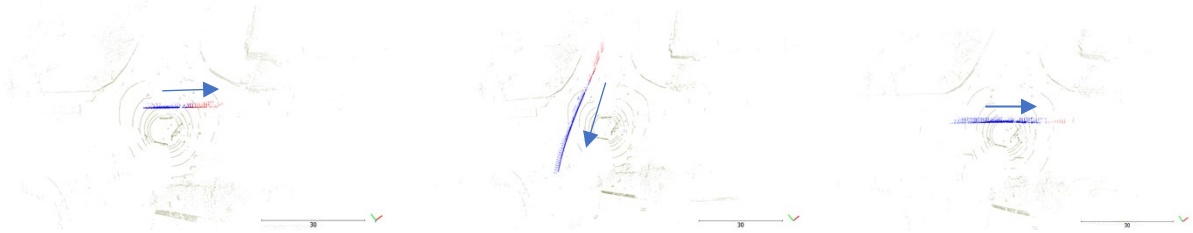
From a qualitative perspective, as shown in Figure 4.13, trajectories from the proposed method are longer than those from the tracking-by-detection method. The differences mainly lie in one (examples 1, 4, 6, 7, 8, 9) or two ends (examples 2, 3, 5) of the trajectories, which is in line with the assumption that low-observable clusters in the far field are highly likely to be absent in the tracking-by-detection method. From a quantitative perspective, seen as the comparison results

in Table 4.11, the proposed method obviously outperforms the tracking-by-detection method: except for examples 2, 3, 4 and 6, N_2/N can be over 90% with several values even reaching 100%. Nevertheless, the highest N_1/N value from the tracking-by-detection method is only 83.9%. The lowest N_1/N value is 60% in example 7, indicating that nearly half of the clusters are missing. Even though this is an extreme example, it indeed happens when the classifier is not properly trained. It has been proven that the proposed method is effective to improve such situation as N_1/N has been increased from 60% to 100% in example 7. The tracking ranges of nine vehicles are shown in Table 4.12, which has further demonstrated the ability of JDAT to increase trajectory ranges.

(a) Vehicle examples 1-3 from Test Site 2



(b) Vehicle examples 4-6 from Test Site 3



(c) Vehicle examples 7-9 froFm Test Site 4

Figure 4.13. The trajectories of nine vehicle examples from two methods: blue is from the tracking-by-detection method and red is from the proposed method.

Table 4.10. Comparison between the tracking-by-detection method and JDAT regarding the range of vehicle trajectories.

Study Sites	Vehicle Examples	Tracking-by-detection			JDAT			Ground Truth			Comparison		
		Start frame	End frame	N_1	Start frame	End frame	N_2	Start frame	End frame	N	N_1/N (%)	N_2/N(%)	N_2/N-N_1/N
2	1	777	849	73	743	849	107	741	849	109	67.0	98.2	31.2
	2	878	967	90	875	976	102	865	991	127	70.9	80.3	9.4
	3	4880	4971	92	4871	4979	109	4863	4996	134	68.7	81.3	12.6
3	4	4221	4274	54	4219	4284	66	4219	4302	84	64.3	78.6	14.3
	5	9110	9163	54	9100	9173	74	9100	9181	82	65.9	90.2	24.3
	6	9105	9162	58	9104	9174	71	9104	9185	82	70.7	86.6	15.9
4	7	0	20	21	0	34	35	0	34	35	60.0	100	40
	8	766	836	71	766	858	93	766	858	93	76.3	100	23.7
	9	926	977	52	926	987	62	926	987	62	83.9	100	16.1
Mean											69.7	90.6	20.9

Table 4.11. Tracking ranges of nine vehicle examples from Tracking-by-detection(D_1) and JDAT(D_2).

Study Sites	Vehicle examples	D_1 (m)	D_2 (m)
2	1	53.26	92.22
	2	48.52	56.81
	3	47.71	57.21
3	4	19.87	26.92
	5	21.15	27.15
	6	17.29	22.16
4	7	11.04	18.68
	8	24.03	31.01
	9	17.41	19.78

4.5.3.2 Continuity of the trajectories

Six vehicle examples, denoted as vehicle examples 10-15 from three study sites (10 and 11 from Study Site 2, 12 and 13 from Study Site 3, 14 and 15 from Study Site 4), are used to justify that the proposed method has the ability to bridge the trajectory gaps caused by miss-detections from the tracking-by-detection method.

In vehicle example 10 at Study Site 2, the trajectory from the tracking-by-detection method (left in Figure 4.14) is chopped into two at the front end due to a short-time occlusion. While the corresponding trajectory from the proposed method (right in Figure 4.14) is successive because clusters that are lost in the tracking-by-detection method are retained on the trajectory. In vehicle example 11 at Study Site 2, the trajectory from the tracking-by-detection method is divided into three parts from the rear end (blue, green and red in the left in Figure 4.15) because some low-observable clusters are missing after vehicle detection. The problem is avoided in the proposed method and a continuous trajectory is generated (right in Figure 4.15).

As to example 12 from Study Site 3 (Figure 4.16), there is a slight occlusion at the beginning (after around 2s when the vehicle started to be tracked) and several affected clusters are overlooked in the detection stage in the tracking-by-detection method, resulting in the interruption of the trajectory. Nevertheless, tracking proceeds smoothly from the beginning to the end in the proposed method. Vehicle example 13 at Study Site 3 is turning right. When the vehicle is being tracked for around 3.5s, the vehicle clusters become too weak to the classifier due to self-occlusion. Thus, tracking is suspended in the tracking-by-detection method until clusters are recovered 0.5s later. As a result, two trajectories turning right are generated, (see left figure in Figure 4.17). While there is no such problem in the proposed method because those low visible clusters are assigned to the trajectory directly in the tracking stage and they do not contribute to the subsequent trajectory classification.

Vehicles 14 and 15 from Study Site 4 are both suffering from occlusions caused by other vehicles. As for vehicle 14, occlusion is severe and the affected clusters only appear to be blurred boundaries. Accordingly, tracking is paused for around 1.5s in the tracking-by-detection method (left in Figure 4.18). While there is no negative influence in the proposed method as can be

concluded from the integral trajectory in Figure 4.18. In terms of vehicle 15 from Study Site 4, despite discontinuous occlusions, tracking is conducted without any resistance in the proposed method. Unfortunately, tracking in the tracking-by-detection method is interrupted twice, generating a trajectory that is cut into three pieces from the rear end (blue, black, and red see right figure in Figure 4.19).

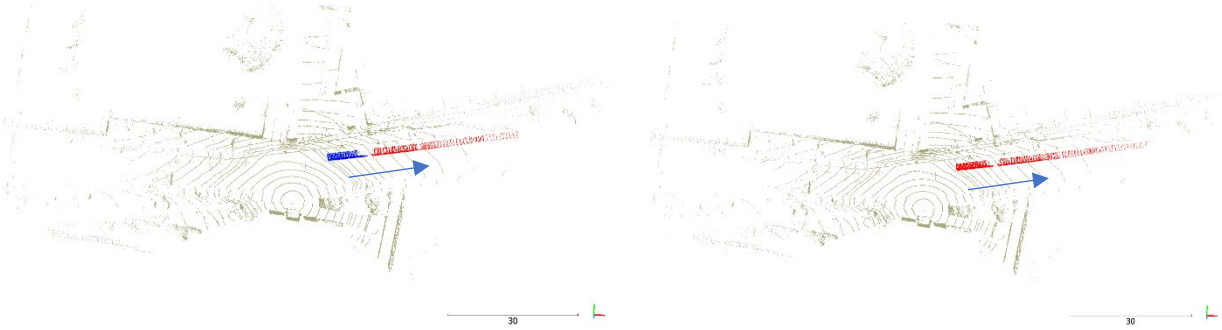


Figure 4.14. Vehicle example 10 from Study Site 2. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

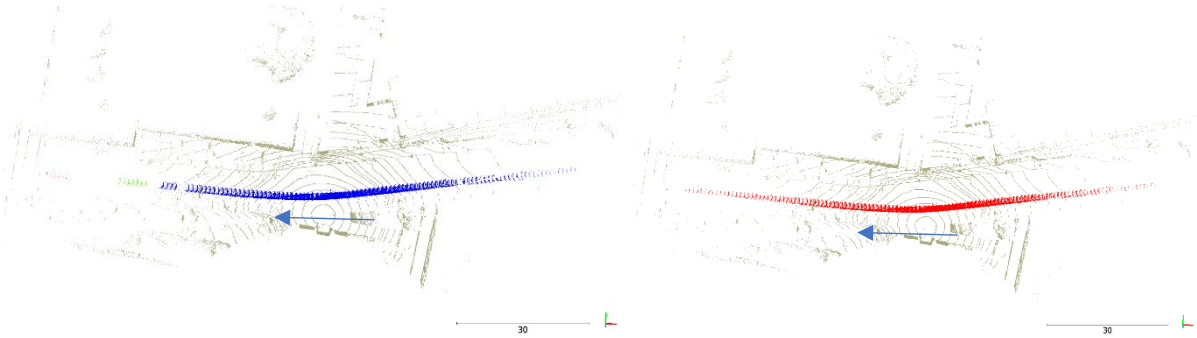


Figure 4.15. Vehicle example 11 from Study Site 2. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

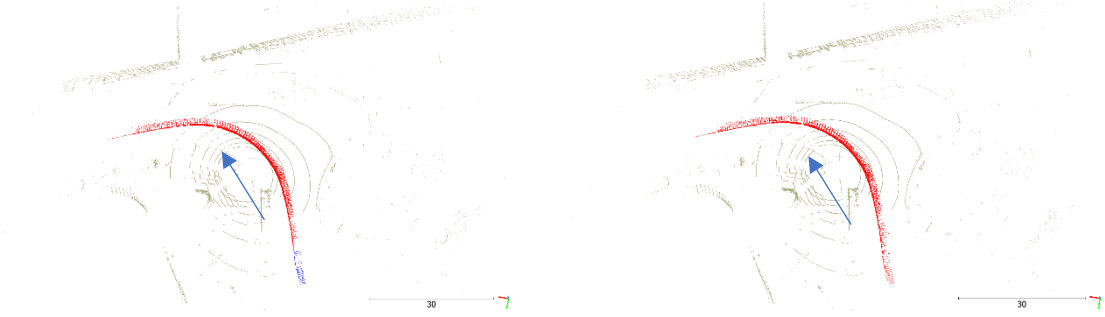


Figure 4.16. Vehicle example 12 from Study Site 3. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

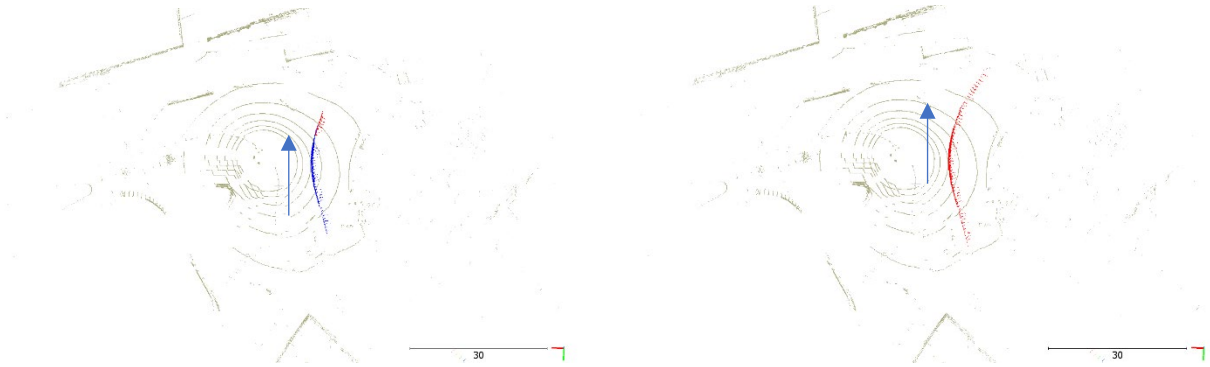


Figure 4.17. Vehicle example 13 from Study Site 3. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

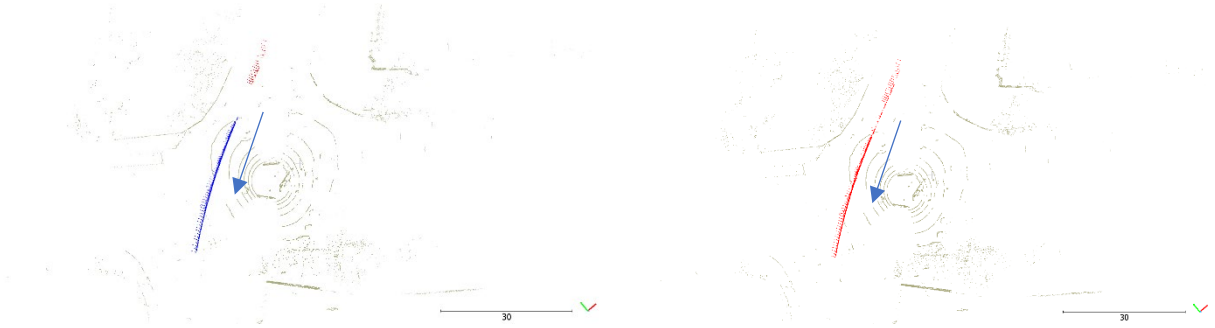


Figure 4.18. Vehicle example 14 from Study Site 4. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

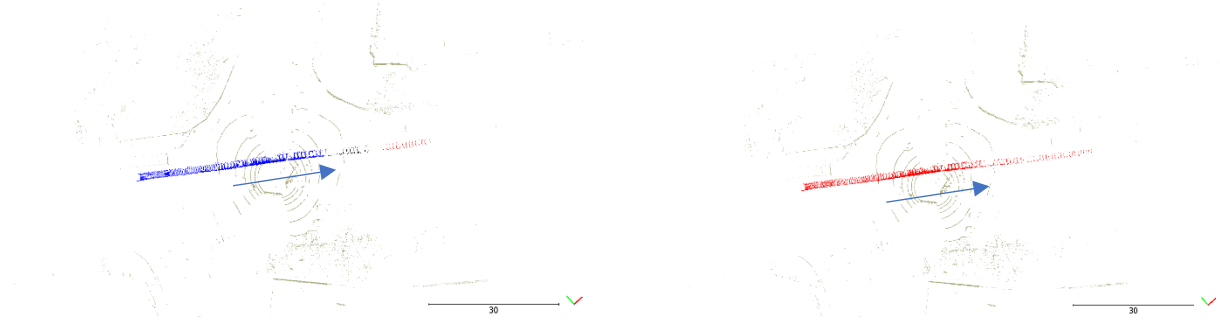


Figure 4.19. Vehicle example 15 from Study Site 4. Left: trajectories from the tracking-by-detection method; right: trajectory from the proposed method.

From the aforementioned comparisons (Sections 4.5.3.1 and 4.5.3.2) based on 15 vehicle examples from three study sites, it can be concluded that moving object trajectories from the JDAT method are of wider ranges than corresponding ones from the tracking-by-detection

method. Besides, the trajectory gaps resulted from the tracking-by-detection method can be stitched by the JDAT method, therefore, the continuity of vehicle trajectories can be improved.

4.5.3.3 The maximum tracking range

As demonstrated above, the proposed method is capable of increasing the ranges of the object trajectories. Therefore, it is reasonable to assume that the tracking range of the lidar sensor can also be increased. The maximum tracking ranges of two lidar sensors at three study sites are measured by the two methods, with statistical results shown in Table 4.13. Figure 4.20 shows the trajectories that are the furthest away from the lidar sensor at three study sites (the one highlighted by a black box).

Observed from Table 4.13, the maximum tracking range at Study Site 2 by RS-LiDAR-32 has been increased from 108m to 111.3m by the proposed method. Even though there is no obvious difference between two methods in the measurements at Study Site 3 and Study Site 4, the results have confirmed that the tracking range by VLP-16 can reach 112.4m. The tracking ranges at Study Site 3 and Study Site 4 are different (112.4m vs 98.2m) despite that the same lidar sensor, VLP-16, was set up. This can be interpreted by the road conditions of two study sites: at Study Site 3, there are no vertical lanes in front of the sensor to the open road, but two such lanes exist at Study Site 4. Therefore, the sensor at Study Site 3 is more likely to 'see' further of the open road than the sensor at Study Site 4.

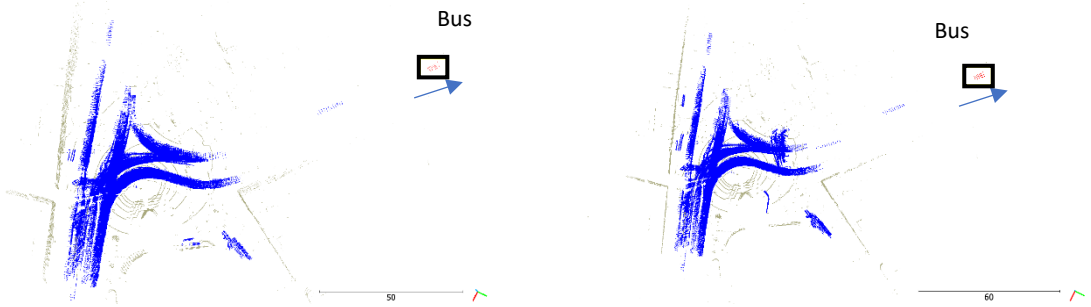
Table 4.12. The maximum tracking range of two lidar sensors at three study sites.

		Study Site 2(RS-LiDAR-32)	Study Site 3 (VLP-16)	Study Site 4 (VLP-16)
Maximum tracking range (m)	Tracking-by detection	108.0	112.4	98.2
	JDAT	111.3	112.4	98.2

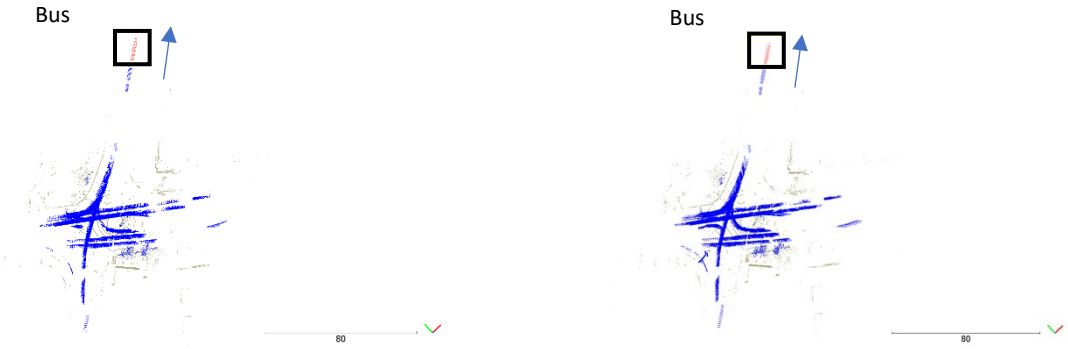
The vehicle that reaches the furthest to the sensor at Study Site 2 are tracked from frame 1478 to frame 1672 in the proposed method (right in Figure 4.20 (a)), while tracking only lasted for a few frames in the tracking-by-detection method before it stopped due to influence of occlusions (left in Figure 4.20 (a)). At Study Site 3 and Study Site 4, the vehicle at the furthest of an open

road branch is a bus (Figure 4.20 (b) and (c)). Buses are higher than normal cars, so they are less likely to be occluded by other vehicles and thus have more points remained on the clusters in the far field. Those far-field clusters are not easily to be overlooked in tracking-by-detection method, therefore, the bus is being tracked until the end of the recording by both methods, resulting in same maximum tracking ranges from two methods.

(a) The furthest trajectories at Study Site 2 from the tracking-by-detection method (left) and the proposed method (right) (Frames 1-1750)



(b) The furthest trajectories at Study Site 3 from the tracking-by-detection method (left) and the proposed method (right) (Frames 1-1000)



(c) The furthest trajectories at Study Site 4 from the tracking-by-detection method (left) and the proposed method (right) (Frames 600-1200)

Figure 4.20. The trajectories furthest to the lidar sensor from two methods at three test sites (highlighted with yellow box).

The algorithm proposed by Wu *et al.* (2018) filters the background by dividing the space into grids with equal size and only considers points within 60m. Therefore, the maximum object detection range can only reach 60m. Based on this background filtering algorithm, vehicles with a max distance of 29.1m from the lidar sensor can be detected and tracked by Wu (2018a). Another background construction algorithm has increased vehicle detection range to 100m (Zhang *et al.*, 2019). The above works are all based on a VLP-16 lidar sensor. In a newly proposed tracking-by-detection procedure (Zhang *et al.*, 2020), the tracking ranges at two test sites with two different lidar sensors, RS-LiDAR-32 and VLP-16, are 45m and 18m, respectively. Compared with the above research, object tracking range in our work has reached 111.3m by RS-LiDAR-32 and 112.4m by VLP-16 tested at three study sites.

4. 6. Vehicle reconstruction and fine-grained classification

Four vehicle reconstruction methods, sequential ICP, sequential NDT, GlobalICP and 2D matching, are validated by a database created from two recordings at Study Site 2. The database includes 20 vehicles with dimensions identified visually through images, among which, No.1 to No.7 are from recording 1 and the others are from recording 2. The clusters of each vehicle within the entire scanning range are associated by the centroid-based tracking procedure in Section 3.4. Those in the far field of the scanning range are prone to inaccurate registration, so vehicle reconstruction is based on clusters in the near field, which means only vehicle clusters with distance to the laser scanner smaller than d are considered in the reconstruction. Four reconstruction methods are both qualitatively and quantitatively evaluated and compared based on vehicles in the validation database.

4.6.1 Visual results of vehicle reconstruction

The point cloud clusters for vehicles in the validation database are retained after the tracking procedure, and then processed by four methods to obtain complete vehicle shapes. The results of four random vehicle examples are shown in Figure 4.21, with each vehicle shape displayed on both the plan-view (left column) and the side-view (right column).

The vehicles are displayed in two views (side and plan) to facilitate the measurement of the length, width and height. Analysed from visual perspective, sequential ICP performs weaker on plan-view as there is an obvious false registration of vehicle 3 (Figure 4.22(c)); sequential NDT and 2D matching perform worse on the side view. For vehicle 1, there is clear error accumulation from 2D matching since the shape is higher than those from other methods (Figure 4.22(a)). Also, there are not enough details such as windows in the shape from 2D matching. For vehicle 2, there is a big distortion near the front wheel in the shape from sequential NDT (Figure 4.22(b)). GlobalICP performs the best without obvious distortions for all the four vehicles.

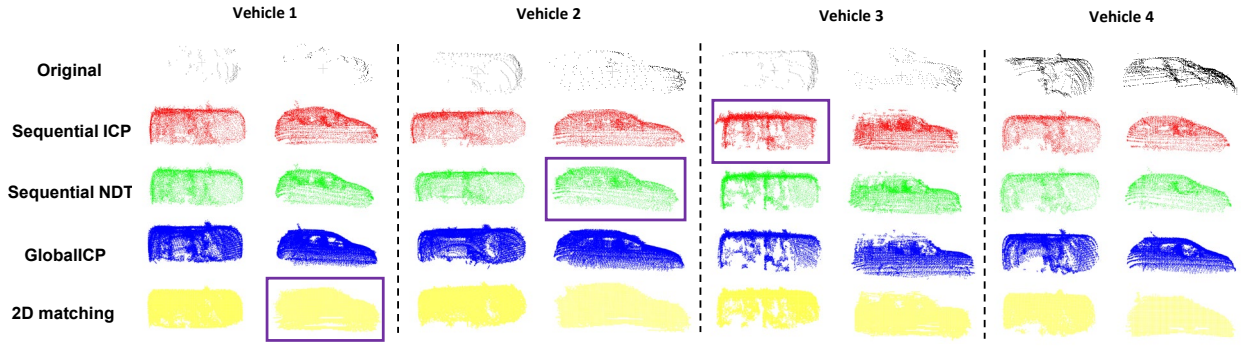
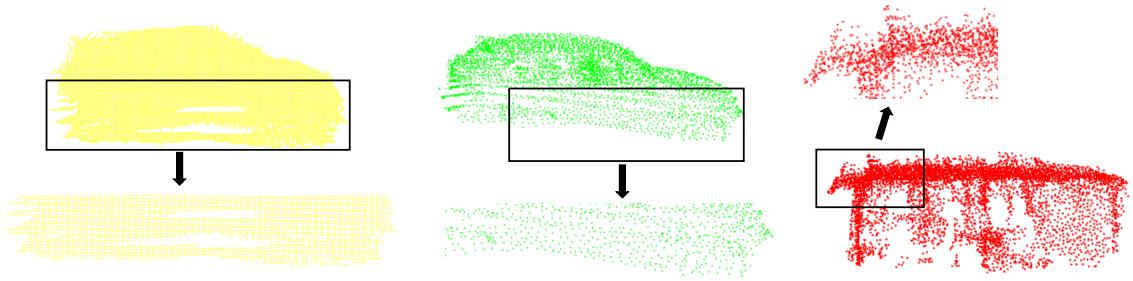


Figure 4.21. Completed shapes of four vehicles by four methods. Each vehicle shape is displayed on both the plan-view (the left column) and the side-view (the right column).



(a) (b) (c)

Figure 4.22. Distorted examples: (a) side view of vehicle 1; (b) side view of vehicle 2; (c) plan view of vehicle 3.

4.6.2 Measurement of reconstructed vehicles

The dimensions (length, width and height) of the reconstructed vehicles are determined by the minimum bounding box in line with the orientation of the vehicle, as shown in Figure 4.23. Table 4.14 shows the comparison between the obtained vehicle dimensions and the corresponding

references. RMSE values regarding three dimensions, denoted as R_L, R_W and R_H, are adopted as the evaluation metrics. To make an overall evaluation, the mean RMSE value of three dimensions is also used as an index, named as Ave_LWH.

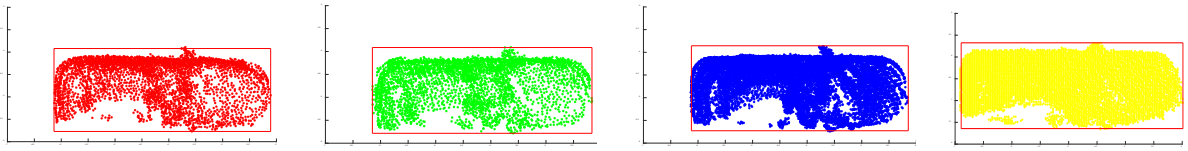


Figure 4.23. The measurements for vehicles constructed from ICP (left), NDT (middle left) and GlobalICP (middle right) and 2D matching (right).

Table 4.13. RMSE of the reconstructed shapes from four methods.

	R_L(m)	R_W(m)	R_H(m)	Ave_LWH(m)
Diff_ICP	0.24	0.17	0.08	0.16
Diff_NDT	0.15	0.16	0.13	0.15
Diff_GlobalICP	0.12	0.13	0.03	0.09
Diff_2D	0.15	0.17	0.15	0.16

It can be seen from the overall statistics that GlobalICP outperforms the other three methods with the smallest Ave_LWH value. It shows more obvious superiority in height calculation with R_H to be only 3cm. NDT performs slightly better than ICP and 2D matching which show equivalent ability.

In our experiments, a series of values ranging from 5m to 10m have been tested in order to determine the optimal value for d. It turned out that the reconstruction results were the best when d equals to 6m.

In addition to vehicle dimension measurement, another application of completed vehicle shapes is to improve the performance of fine-grained vehicle classification. Unfortunately, according to the observed data, it is difficult to create a decent training dataset containing sufficient complete vehicles of different categories, especially trucks and buses. Since the lidar data covers urban areas, the number of trucks is quite small. Besides, although there is a relatively larger number of buses in the original data, it is impossible to generate enough high-quality complete shapes due to lack of details on the top of buses.

4.6.3 Fine-grained vehicle classification

As explained above, it is unrealistic to implement fine-grained vehicle classification on complete vehicle shapes. Therefore, it is still performed on vehicle clusters, specifically, the representatives of the vehicle trajectories. The feature set exploited to train the classifier is the same as that for newly trained RF in vehicle detection, whereas with a different ranking of sub-feature importance, seen as Figure 4.24.

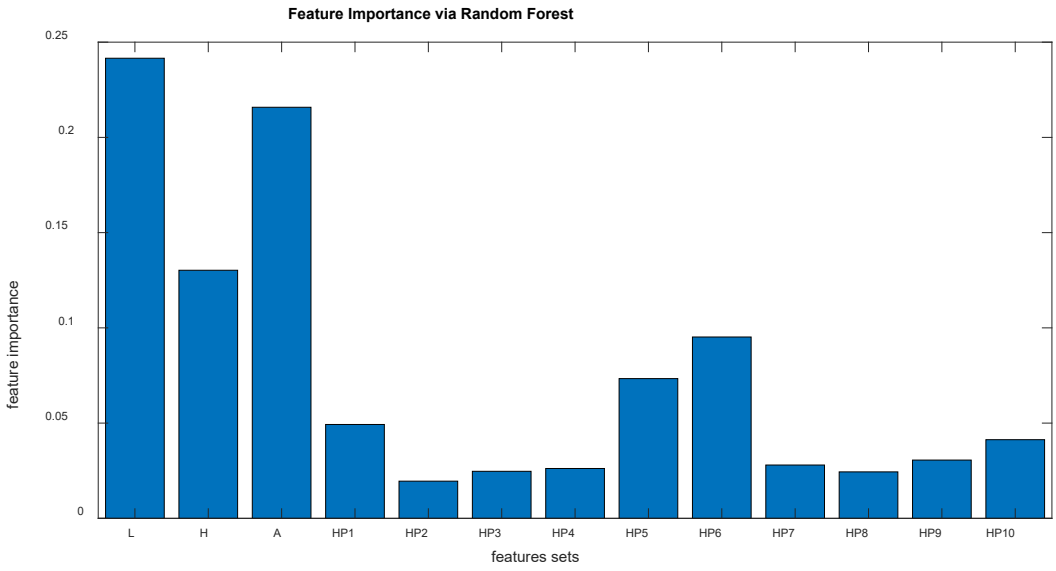


Figure 4.24. The importance of the sub-features.

A dataset including 551 car clusters, 423 van clusters, 93 truck clusters and 268 bus clusters was built based on lidar data from Study Site 1 and Study Site 2 (some samples are shown in Figure 4.25). The dataset is split into three subsets, 0.6 of it for training, 0.25 of it for validation and the remaining for testing.

(a) (b) (c) (d)

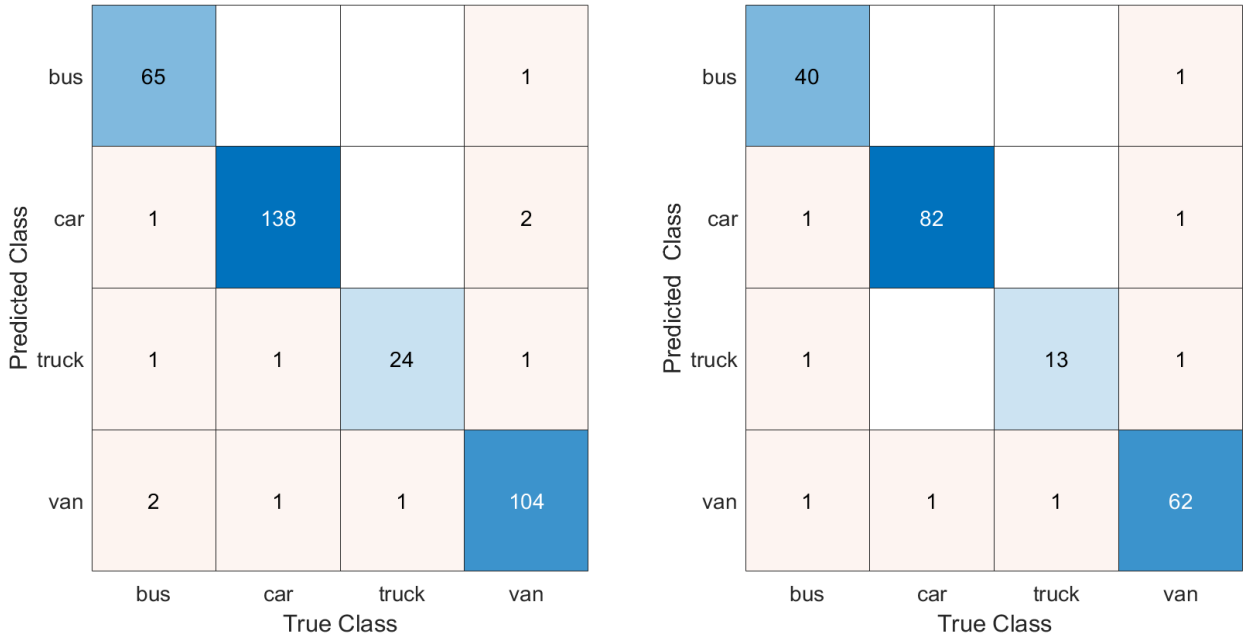
Figure 4.25. Samples of each vehicle category: (a) car; (b) van; (c) truck; (d) bus.

The confusion matrices on both validation and test subsets are shown in Figure 4.26 and corresponding statistics are displayed in Table 4.15. Overall, classification performance is

promising with macro F_1 values of four classes on both subsets to be over 0.95. It can be seen from the results of the test set that truck is the most difficult to be distinguished because F_1 score is the lowest. One possible reason is that the number of training samples of truck is smaller than those of the other three categories. However, it is impossible to achieve any improvements currently due to data limitation.

Table 4.14. Performance of classifier on validation and test subsets.

	bus			car			truck			van			Macro F_1
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	
Validation	0.98	0.96	0.97	0.97	0.99	0.98	0.96	0.96	0.96	0.95	0.95	0.95	0.97
Test	0.98	0.95	0.96	0.96	0.98	0.97	0.87	0.93	0.90	0.95	0.94	0.95	0.95



(a) (b)

Figure 4.26. Confusion matrix of (a) validation set and (b) test set.

4.7 Discussion

Based on the experiments and results, details or potential of the developed traffic monitoring system are further discussed in this section, considering real-world applications.

4.7.1 Traditional vehicle classifiers

After static point removal, in the initial stage of this research, vehicle detection was simplified to a vehicle and non-vehicle classification problem and a simple rule-based classifier and two machine learning-based classifiers, SVM and RF, were realised. The training dataset was sufficient for a classifier intended to distinguish vehicles from other moving objects despite the small number of samples. On one hand, in urban environments, "other moving objects" mainly refer to pedestrians, cyclists, motorcyclists, and a few number of false alarms. Since their features are rather different to that of vehicles, they can be distinguished easily from vehicles. Therefore, the resulting precision and recall were relatively high and this step should not be considered as a bottleneck for future experiments at early stage. The rule-based approach produced higher recall and was hence adopted for the vehicle tracking and high accuracy speed estimation framework.

With more lidar data available, another RF was trained with more distinguishable feature sets. Rather than vehicle and non-vehicle binary classification, the new classifier aims to classify moving objects into three classes, vehicles, pedestrians and others, which provides opportunities for wider traffic applications. For example, it is possible to predict pedestrians' crossing intention and thus avoid the collision between vehicles and pedestrians once their trajectories are obtained in the later tracking procedure. One common road user category in urban city environments, 'cyclist', should ideally be regarded as a single class. However, it is currently ignored due to the small number in the available lidar data. It is necessary to treat cyclists separately when more data available. In terms of 'others', even though there is no such class in KITTI vision benchmark suite (Geiger *et al.*, 2013), it is regarded as a single class in this research since there are some moving clusters from trees or moving bushes that could not be discarded after clustering. However, the classification results for 'others' through RF are much poorer than the other two classes due to insuffient training samples.

4.7.2 PV-RCNN vehicle detector

The advanced 3D object detection network, PV-RCNN, has been applied in this research. It outperformed the reported results in the original work (Shi *et al.*, 2020a) with large margins. The

performance of pedestrian was worse than vehicle according to the inference that the limited number of key-points may harm the performance of objects with small sizes (Shi *et al.*, 2020a), which is also the reason why enlarging the number of training samples of pedestrians by adding KITTI data did not provide any improvement. Vehicles were firstly detected by PV-RCNN and later fine-grained classified into different categories by the RF classifier, with the consideration that discriminating vehicles from other objects first and further classifying them into different categories can usually provide better performances. It would be interesting to apply PV-RCNN as a multi-class detector when more training data is obtained. Moreover, PV-RCNN was used as an object detector in the way it was originally presented. It is more interesting and efficient to use it as a classifier in this work because moving object clusters can be easily extracted beforehand through the first two steps in the three-step vehicle detection workflow. Further trials aim at adapting the network to make it directly operate on object proposals. PV-RCNN was operated on moving points and it has also been tested on original lidar data to provide comprehensive comparisons. There should also be a comparison between PV-RCNN and RF at later stage.

One important threshold in the evaluation metrics is IoU. According to Shi *et al.* (2020a), 0.7 is set for vehicle, 0.5 is set for pedestrian. While in the experiments of this research, 0.5 is set for both classes. Trials have been made to adjust the values but resulted in very minor changes in the results.

4.7.3 Transferability

The above 3D object classifiers and detector were trained from data collected from the first three study sites. The dataset for initial SVM and RF was from Study Site 1 and Study Site 2, but it has been demonstrated that the two classifiers could be adapted to Study Site 3 very well. The new RF was trained from data collected from Study Site 1 to Study Site 3. It also delivered good performance on Study Site PV-RCNN was trained by data from Study Site 1 and Study Site 2, and only tested at these two sites at present. Further tests on Study Site 3 and Study Site 4 are needed to validate the transferability.

4.7.4 The vehicle tracking and high accuracy speed estimation framework

As for vehicle tracking and high accuracy speed estimation, several parameters in the centroid-based tracking stage are important, as shown in Table 4.7. One of these parameters is the initialization threshold. If the association probability of a detection within the assignment gate is lower than the threshold, a new track will be generated. This parameter is critical to decide if a track should end when severe occlusion appears. For example, in heavy traffic flow, if the vehicle being tracked is completely occluded and consequently reobserved with an association probability lower than the initialization threshold, a new ID will be assigned to the subsequent detections. Thereafter, tracking refinement will resume for the new ID. Whereas, in light occlusion, in which the vehicle being tracked is partially occluded for a short period, some of the clusters will be incomplete and lower association probability may arise. To keep tracking continuous in this situation, a small value of 0.1 is assigned to the initialization threshold. In subsequent tracking refinement, if one image of a pair is partly affected, matching can still be conducted between them. Issues may arise that the matching accuracy is low and the estimated speed could be noisy. To tackle this issue, a smoothing algorithm is utilized in the final stage to filter out noise in the speed values.

The performance of the centroid-based tracking mainly relies on the vehicle detection results. In the framework, the tracking accuracy was not quantitatively assessed as all detected vehicles were correctly tracked at both Study Site 2 and Study Site 3, based on visual inspection. This mainly results from the high lidar sensor frame rate (0.1s), which implies that clusters of a vehicle in two successive frames can be easily associated due to the small spatial distance between them. False tracks are mainly caused by false positives (non-vehicle road users, trees, or similar objects) in the detection stage. Tree like objects can be easily filtered by the length of the trajectories since they are almost stationary throughout the tracking period. However, some non-vehicle road users, such as pedestrians walking closely together along the road, cannot easily be discarded. According to Zhang *et al.* (2020), a potential improvement to enhance the detection accuracy could be integrating semantic constraints, such as extracting road boundaries beforehand to exclude pedestrians. As a new RF classifier has been trained based on more distinguishable feature sets and could provide a 91% detection accuracy for pedestrians, the

aforementioned false tracks can be easily removed from the detection stage. Tree like trajectories that could not be removed by trajectory length can be singled out by the new RF since it is a three-class classifier.

The estimated speeds from the first framework are validated against a reference system that is considered to provide a higher order of accuracy. The RMSE of the reference data was about one-tenth of that of the lidar data. Speed validation was also conducted by a test vehicle with an on-board diagnostics logger recently by Zhao *et al.* (2019). The reported average absolute speed difference between speeds from lidar data and reference data, which is equivalent to MAE in our work, is as high as 0.639m/s. In comparison, the average MAE of all the cases in our work is 0.18m/s. A more accurate reference system and a high accuracy speed estimation framework allowed full exploration of the capacity of lidar speed estimation.

4.7.5 The JDAT framework

Nine vehicle examples from three study sites have been used to show that the proposed JDAT framework has improved the object trajectories by increasing their lengths. Another six examples from three study sites have justified that the proposed framework has also improved the continuity of the object trajectories. The effectiveness has been assessed through both the qualitative and the quantitative analysis of various examples from different traffic scenes. Point cloud data from three study sites has been processed to demonstrate that the proposed method has enlarged the object tracking range of two commonly used lidar sensors compared with the tracking-by-detection method. As can be seen from the experiments, the difference at Study Site 2 was obvious (108m vs 111.3m), which can be explained by the ability of the proposed method to keep the continuity of the object trajectory under light occlusions and reduced points. Even though there was no obvious difference in the measurements from two methods at Study Site 3 and Study Site 4, the results have confirmed that the maximum tracking range of a VLP-16 lidar sensor is 112.4m, which is important guidance for real-world lidar sensing applications.

In addition to the proposed JDAT framework, a STC scheme is introduced in 4.4.2 to alleviate the negative influence that tracking is suffering from detection in traditional tracking-by-detection method. In STC, three procedures, segmentation, tracking and classification, are conducted in

sequence, while vehicle detection and tracking are conducted as two parallel diagrams in the proposed framework. PV-RCNN, as an object detector, cannot be directly applied to individual representatives that are selected from object trajectories as the way RF is used in STC. Efforts have been made to place the representatives from all the obtained trajectories into one uniform frame and send it to PV-RCNN. Unfortunately, the results were not satisfactory. A better solution is to perform PV-RCNN on original lidar frame (or the frame that only contains moving points as in this research), and then locate the representatives in the detection results to determine the categories of them. It turned out that the proposed method provided better classification results than STC. However, the results just mean the proposed method performs better than STC in terms of the whole framework. Ablation studies are needed when PV-RCNN is adapted to a classifier at a later stage when it will be worth replacing RF with PV-RCNN in STC scheme to make fairer comparisons.

The input of the JDAT framework is original lidar data and the output are trajectories of vehicles and pedestrians. Detection and tracking are performed in parallel in the framework. In a similar work where joint object detection and tracking is also performed (Huang and Hao, 2021), the realization of parallelism relies on an object detection network and an object correlation network. The object correlation network is only part of the object tracking procedure, which means detection and tracking in this work are not totally in parallel. Although object tracking in the JDAT framework is not based on advanced deep learning strategies, it is totally independent from object detection, which makes it more flexible and capable of generating higher quality outcomes such as trajectories with wider range and better continuity.

4.7.6 Vehicle reconstruction and fine-grained classification

In the current reconstruction procedure, only clusters that with distance to the laser scanner smaller than *d* are considered. *d* should be adjusted according to the traffic conditions and the location where the laser scanner is installed. Currently, the algorithm has only been tested at Study Site 2 with d set to 6m. Further tests on other study sites are needed especially at Study Site 3 and Study Site 4 where a different laser scanner, VLP-16, is installed. An alternative way is to perform reconstruction on the representatives of the trajectory as they maintain predominant

features of the vehicle. One issue that should be addressed is occlusion. If occlusion appears in the near field where reconstruction is conducted, the performance would be affected due to defective clusters irrespective of the way clusters are chosen as input into the reconstruction.

In the first way, the defected clusters would be used in reconstruction, resulting in distortion. In the second way, the defected clusters might not be used in reconstruction because of small sizes or small areas. However, as a result, the exploited clusters are not consecutive, which might cause registration error. Fortunately, GlobalICP has better ability than the other three methods to deal with this issue as the least squares adjustment strategy is adopted to estimate the overall transformation parameters.

Shape completion on lidar data is generally conducted with supervised strategies in which paired training data is needed to learn generative models. Following the paradigm of PCN (Yuan *et al.*, 2018), point clouds can be completed with neural networks trained from artificial CAD in graphics datasets, such as ShapeNet (Chang *et al.*, 2015), VPC-Net (Xia *et al.*, 2021) and S2U-Net (Xia *et al.*, 2020a). However, it is expensive to acquire CAD models and they are limited to some categories. Considering the mentioned drawbacks, the proposed method in this thesis could utilize temporal lidar data to complete vehicle shapes in an unsupervised manner. Completion starts once the vehicle under tracking enters the reconstruction zone, thus, it can be further integrated into a real-time traffic monitoring system without affecting the efficiency.

The completed vehicle shapes were evaluated in a macro perspective by a validation database containing 20 vehicles with specific dimensions. The estimated values for length, width and height of the completed shapes were compared with corresponding values from the validation database. While the evaluation metrics adopted in state-of-the-art methods are Chamfer Distance[3] and Earth Mover's Distance[4] between the completed point cloud and ground truth which are calculated on point level (Xia *et al.*, 2020a), in a micro perspective. Through the evaluation method in the work of this thesis, not only the completion methods can be validated, but also vehicle dimensions can be obtained.

[3] Chamfer Distance (CD) is an evaluation metric for two point clouds. For each point in each cloud, CD finds the nearest point in the other point set, and sums the squares of distances up.

[4] Earth Mover's Distance is a measure of the distance between two probability distributions over a region *D.*

As for fine-grained vehicle classification, the principal drawback is the low accuracy of 'truck' caused by small number of training samples. It is impossible to make obvious improvement based on the current data. As there are 488 trucks in KITTI data, it might be helpful to add them into the training dataset.

4.8 Summary

In this chapter, the lidar sensors employed in this research were introduced in Section 4.1, followed by the description of the study sites in Section 4.2. For vehicle detection (Section 4.3), the performance of several classifiers in initial trials, a newly trained RF and PV-RCNN has been demonstrated, respectively. Particularly, a comparison was conducted between PV-RCNN on original lidar points and moving points. The advantages of two vehicle tracking frameworks have been evaluated by various case studies in Section 4.4 and 4.5. Three sets of vehicle speeds, centroid-based speeds, refined speeds and the reference, were compared in Section 4.4 to thoroughly confirm the ability of the vehicle tracking and potential to create a more accurate speed estimation framework. Both a tracking-by-detection method and a STC scheme were compared with the proposed JDAT framework in Section 4.5. Vehicle reconstruction and fine-grained classification were introduced in Section 4.6. Following the experiments and results, there is a comprehensive discussion section with regard to each element of the developed system to further explore the possibilities of this research. Inspired by the developed system, further discussion on real-world lidar-based traffic monitoring is conducted in Chapter 5.

Chapter 5. Discussion

Based on the study in this thesis, further discussion on lidar-based traffic monitoring is carried out, to provide insights and guidance for real-world implementations. Influential factors related to built-in features of lidar sensors**,** installation strategies for real-world applications and several external aspects that have impact on lidar data processing are discussed in this chapter.

5.1. Influence from built-in features of lidar sensors

As shown in Table 4.1, built-in features of lidar sensors mainly include the number of laser beams, horizontal and vertical FOV, horizontal and vertical angular resolution, accuracy of point clouds, et.al. The number of laser beams matters greatly in traffic monitoring operations because the point density would vary to a large extent. Laser beams are vertically distributed based on the vertical angular resolution, either evenly or non-linearly, resulting in different numbers of points from the same object with different distances to the lidar sensor. The high accuracy 3D information of lidar data makes it easier to obtain high accuracy trajectory-level traffic data. However, inherent lack of colour information of Red, Green and Blue (RGB) colour information makes the identification of objects less straightforward than using video imagery

5.1.1 Number of laser beams

In traffic monitoring, the number of lidar points collected from road users is limited since the majority of the lidar data are unrelated background points. In this regard, road users should be scanned by as many laser beams as possible so that sufficient points could be obtained to facilitate further operations such as vehicle detection, reconstruction and classification which relate to Objectives 1, 3, and 4 of this research.

High-profile lidar sensors with 64 beams or 128 beams are still too expensive to deploy as mainstream sensors for infrastructure-based traffic monitoring (Zhao *et al.* 2020) . Therefore, two lower-profile lidar sensors, VLP-16 (16 beams) and RS-LiDAR-32 (32 beams) were employed in this research. Other lidar sensors such as Ultra Puck (VLP-32C) or HDL-64ES2 from Velodyne

can also be tested when available (Zhao *et al.* 2020). A suggested experimental design is as follows:

Three lidar sensors of different number of laser beams, HDL-64ES2, VLP-32C (or RS-LiDAR-32) and VLP-16, are adopted to collect lidar data in the experiment. The objective of this experiment is to quantitatively show the influence of the number of laser beams. Vehicle detection, vehicle reconstruction and classification will be realised based on lidar data from three sensors.

Data collection

At the test site, there are two ways to install the three lidar sensors: (a) if there are more than four people involved in data collection, three lidar sensors can be installed at the same time. A camera is recommended to install near the lidar sensors; (b) if there are less than four people for data collection, three lidar sensors are suggested to be installed one at a time. Again, a camera is recommended to install near the lidar sensor. The recording duration for each lidar senor will be determined by the traffic situation to achieve an appropriate number of vehicle observations to train the algorithms.

Data processing

Vehicle detection, vehicle reconstruction and vehicle classification will be performed three times based on data from three sensors. More data will be collected until sufficient data to construct a training dataset for vehicle detection or vehicle classification is achieved.

Result analysis and comparison

Comprehensive analysis of the results will be conducted when the processing is completed. Comparisons among three sets of results will be made before conclusions are drawn.

5.1.2 Distance to the lidar sensor

The density of lidar point clouds changes with the distance to the lidar sensor. If an object is located near the lidar, intensive data points are collected and thus present a finely resolved description of the object. If an object is far away from the lidar, only sparse data points are collected, particularly for the small-sized objects. This is the motivation for developing the JDAT framework in this study, which relates to Objective 2 of this research. Besides, to use the best

data for better accuracy, vehicle reconstruction, the method to realise Objective 3 of this research, is only conducted in the near field. It is worth exploring some new algorithms which take distances to the lidar sensor into account, to improve procedures such as clustering (e.g. , such as a modified DBSCAN clustering algorithm proposed by Zhao *et al*. (2019). Minimal Points (MinPts) and searching radius (ε) are the two primary parameters of the traditional DBSCAN algorithm (Ester *et al*., 1996). If the number of data points within a searching area is greater than or equal to a predefined MinPts value, those data points will be clustered to form an object. Because of the unique features of roadside LiDAR data, it is difficult to obtain accurate clustering results by using fixed MinPts and searching radius. Therefore, the modified DBSCAN clustering algorithm effectively adjusts the MinPts and searching radius at different locations. The detailed process can be referred to the work from Zhao *et al*.

5.1.3 Adjustable horizontal FOV

The default value of the horizontal FOV of lidar sensors is recognized as 360° according to the configurations in the manual. It is noteworthy that this attribute is adjustable even though the default value was used in the current work. For example, according to the manual of VLP-32C (Velodyne, 2016), the horizontal FOV can be customized by setting values to 'FOV Start' and 'FOV End' in the configuration screen. Inspired by this guidance, the horizontal FOV of the lidar sensor in roadside traffic monitoring systems could be adjusted according to the road conditions in order to reduce the amount of unrelated data. To be specific, when collecting data at straight roads, the horizontal FOV is set to 180° facing the road; when collecting data at intersections where the lidar sensor is deployed at the corner, the horizontal FOV is set to 270°, covering the intersection; when the lidar sensor is installed at a roundabout, the horizontal FOV is set to 360°. For large-scale real-world traffic monitoring, the above suggestion is useful because reducing the amount of unrelated data is necessary. A robust experimental design is needed based on this suggestion and according to the road conditions at the test site. For example: when collecting data at a straight road section, the lidar sensor is installed at a known distance to the road edge, and the horizontal FOV is set to 180°. Whilst looking through the lidar viewer the location of lidar to the

road edge could then be adjusted to make sure the complete road is covered. When all the preparations are completed, data collection could then be officially started.

5.1.4 High accuracy 3D information

Another feature of lidar sensors is that they capture the environments with 3D points. The measurement accuracy of these points can be as high as 3cm (Velodyne, 2016). On the basis of the high accuracy 3D information, higher quality traffic data such as higher accuracy vehicle speeds, part of what Objective 2 expects, could be obtained according to the study in this thesis. Whereas, it cannot be guaranteed that equivalent performance could be achieved by video data, as seen from the comparison in Section 4.4.5.

5.1.5 Lack of textural information

Although lidar has shown superiority in traffic monitoring, there are inherent disadvantages in lidar points such as sparsity, irregularity and lack of textural information. This study has shown that lack of RGB information is a drawback of lidar data in traffic monitoring. It has negative effect on vehicle reconstruction and vehicle classification, which relate back to Objective 3 and Objective 4 of this research: i) the validation database for vehicle reconstruction was constructed with the aid of video imagery to identify the make and model of the target vehicles; ii) The database for fine-grained vehicle classification was created with the help of video imagery to identify vehicles whose categories could not be identified from lidar data

5.2. Suggestions for real-world lidar installation

In agreement with Zhao *et al.* (2020), location, height and inclination are three important factors in real-world lidar installation. In addition to them, installation of multiple lidar sensors and installation for specific purpose are also illustrated in the following sections:

5.2.1 Location

Lidar sensors can be temporarily installed on a tripod for short-term data collection or permanently mounted on roadside infrastructures for long-term data collection. The first

strategy was adopted in this research, in which the distance d between the tripod and the road edge is pivotal.

When the height of the lidar sensor (horizontally installed) is fixed, the undetectable range is also fixed (seen as Figure 5.1, the undetectable range is the maximum horizontal distance between the lowest laser beam and lidar's central axis). If the lidar sensor is too close to the road edge, more road region is located in the undetectable range, which might cause some road users to be missed. On the contrary, if the lidar sensor is too far from the road edge, some laser beams may shoot outside of the road and there might not be enough points collected from the road region. In this research, d was determined at each study site by visual observation from the lidar viewer (VeloView and RSView).

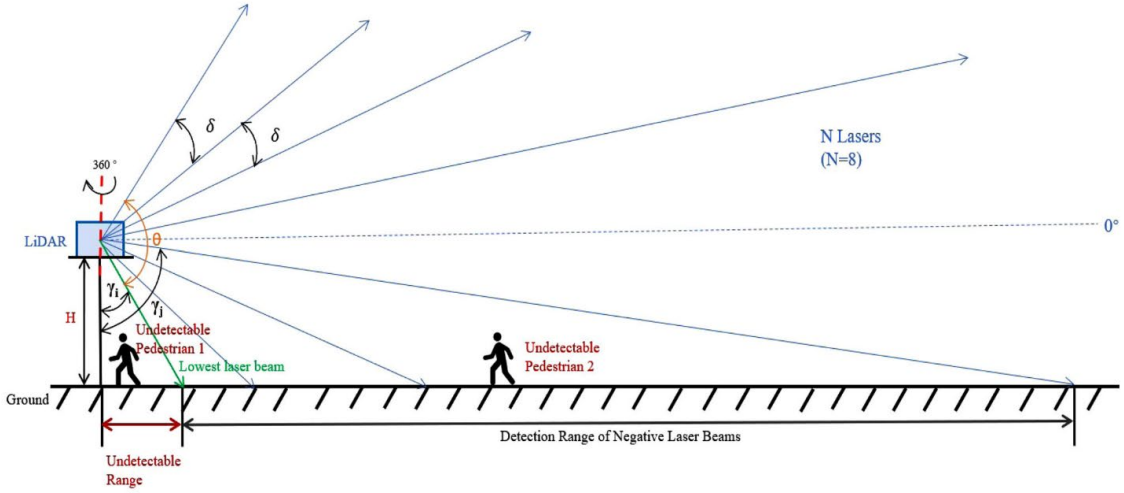


Figure 5.1. Undetectable range of the lidar sensor (Zhao et al., 2020)

5.2.2 Height

Height of the installation refers to the vertical distance between the lidar sensor and the ground surface. How far the laser beams can reach is determined by the height of installation and the built-in features of the lidar sensor, as well as the location and height of the target object.

Once the distance between the tripod and the road edge d is settled, installing the lidar sensor at a higher position can minimize the effect of occlusion. However, the undetectable region may become larger if the sensor is installed too high. In this research, lidar sensors are erected at 1.8

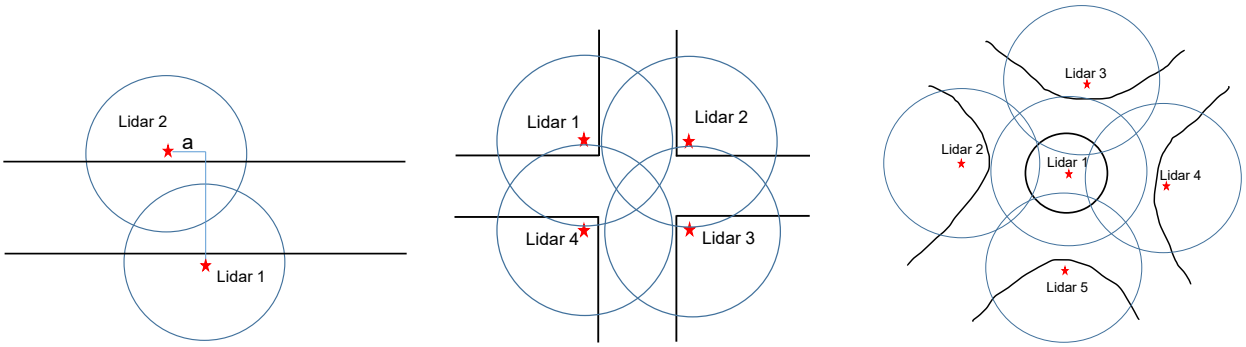
metres, higher than cars, cyclists and pedestrians and lower than large vehicles such as vans and buses.

5.2.3 Inclination

Ideally, the lidar sensor should be installed on a horizontal plane. Since all the laser beams are rotated along the sensor's central axis forming conical surfaces, target objects with the same distance to the sensor but in different orientations are scanned identically because the angle of each laser beam relative to the horizontal plane is fixed. However, in practice, there is inevitably an inclination either because the lidar sensor is not totally levelled or because the ground surface is not horizontal. With this inclination, target objects with the same distance to the lidar sensor but in different orientations are scanned differently. If the difference is obvious, operations to adjust the data to a horizontal plane must be conducted. Therefore, it is important to install the lidar sensor as horizontal as possible in real-world applications to avoid extra workload in data processing.

5.2.4 Multiple lidar sensors

Occlusions from other road users are unavoidable if only one lidar sensor is used even though the problem can be minimized by increasing the height of the sensor. Also, there is no evidence that the proposed JDAT framework is capable of dealing with heavy occlusion. Hence, multiple lidar sensors are recommended, if available. Two corresponding sensors should be located at different sides of the road, with certain distance (a in Figure 5.2(a)) along the road. The value of a should be determined by considering the effective tracking ranges of the lidar sensors provided by this research. At a crossing, one lidar sensor at each corner is expected in order to provide enough overlapped data for each arm of the junction (Figure 5.2(b)). At a roundabout, a lidar sensor is needed in the central region in addition to those at the corners to make sure there is overlapped data for objects from all directions (Figure 5.2(c)).



(a) A road section (b) A crossing (c) A roundabout

Figure 5.2. Installation of multiple lidar sensors.

5.3. External factors

5.3.1 Study Sites

There are four study sites in the city employed in this research: around a corner on a road, a straight road section, a road intersection and a roundabout. Even though these sites cover both busy and quiet traffic conditions, the number of them is still small and they are only constrained to urban areas. For further comprehensive research, more study sites inlcuding those on highways and in suburbs (to cover more vehicle categories), both in the UK and beyond, are required.

5.3.2 Weather conditions

Wojtanowski *et al.* (2014) found that lidar is susceptible to adverse weather conditions such as rain, fog, snow and wind. Therefore, investigating how to improve the accuracy of traffic data under adverse weather is significant for traffic applications.

The current data processing algorithms for roadside lidar are usually developed assuming normal weather conditions. For example, all the lidar data used in the study contained in this thesis was collected on sunny days. Adverse weather could be challenging for data processing especially for road user detection including background filtering and object clustering (Wu *et al.*, 2020). For example, the method proposed by Zhao *et al.* (2019) could not cluster the points correctly under snowy weather. Jokela *et al.* (2019) found that lidar sensors' performance decreased with the increasing density of fog and the distance between the target and the lidar. It is still necessary to

quantitatively analyse the influence of different adverse weather conditions on roadside lidar and to adapt the methods to accommodate such conditions.

5.4. Summary

In this chapter, several built-in features of lidar sensors (number of laser beams, distance from the road to the lidar sensor, adjustable horizontal FOV, higher accuracy 3D information and lack of textural information) that need to be considered in real-world applications have been estimated and suggestions on further study or application related to each factor are given. External factors including study sites and weather conditions have also been mentioned in this chapter. For real-world lidar installation, suggestions on location, height, inclination as well as the use of multiple lidar sensors are provided, all of which are helpful to practitioners. Comprehensive conclusions from this research by revisiting the research aims and objectives, research contributions, summary and further work are made in Chapter 6.

Chapter 6. Conclusions

In this chapter, the aims and objectives of this research are revisited, and the contributions of this research are clarified. Conclusions are drawn and anticipated work that can improve this research is also summarized.

6.1 Revisit research aims and objectives

The research reported in this thesis aimed to develop an integrated lidar-based roadside traffic monitoring system that can provide fundamental traffic information, including the number of vehicles, vehicle dynamics, vehicle dimensions and vehicle types. The overall aim of the PhD project was achieved via fulfilment of the following objectives.

Quantification of vehicle numbers by vehicle detection

As illustrated in Sections 3.1, 3.2 and 4.3, this objective has been achieved through two methods: i) a three-step detection method in which traditional machine learning classifiers SVM and RF were used in the third step to distinguish vehicles and non-vehicles; ii) a deep learning method where PV-RCNN was adopted as the vehicle detector. In the first method, SVM and RF were initially trained using a small amount of data, resulting in an overall accuracy higher than 0.92, which is competitive or even better than most of the existing traditional machine learning-based 3D vehicle detection methods. A three-class RF classifier was trained with a larger dataset and more distinguishable feature sets when more data was available, resulting in a F_1 score of vehicle and pedestrian of around 0.9. Macro F_1 of the three classes was 0.85, with high possibility to be increased when more samples are available for the third class. In the other method, an advanced 3D object detection network PV-RCNN was exploited attempting to improve vehicle detection performance. The results outperformed the reported accuracy from the original work with obvious margins: AP values for vehicle at two study sites (96.6 and 85.0) are higher than those for either easy (90.3) or moderate (81.4) cars from the original work, despite the fact that vehicles in our work include various categories. AP values for pedestrians (78.7 and 54.2) are higher than the reported accuracy for easy pedestrians (52.2).

Acquisition of high-quality vehicle trajectories through vehicle tracking

According to Sections 3.4, 3.5, 4.4 and 4.5, this objective has been addressed by two vehicle tracking frameworks developed from two popular strategies, i.e. tracking-by-detection and joint vehicle detection and tracking. The first framework intends to obtain higher accuracy vehicle speeds by introducing a tracking refinement module in addition to the centroid-based tracking procedure. An individual speed reference system is utilized to validate the estimated vehicle speeds. It has been proven that this framework can provide a mean speed accuracy of 0.2m/s, with an improvement of 46.3% compared with centroid-based tracking. The other is a joint vehicle detection and tracking framework in which tracking, and detection are conducted in parallel so that misdetections from the vehicle detection stage can be mitigated. The average range of vehicle trajectories are increased by c. 21% from the results of different scenes. The continuity of the trajectories is also enhanced and the maximum effective tracking ranges of both tested laser scanners in different traffic scenes are found to exceed 110m.

Measurement of vehicle dimensions through vehicle reconstruction

As can be seen from Sections 3.6 and 4.6, this objective has been fulfilled by implementing vehicle reconstruction from the perspectives of both 2D and 3D (four methods in total) on the tracking results from Objective (2). The measurement of vehicle dimensions is accordingly realised from the completed vehicle shapes. The obtained vehicle dimensions are assessed by a validation database with a RMSE value smaller than 0.16m on the average of length, width and height measurements.

Identification of vehicle type through fine-grained vehicle classification

According to the data we possess, it is difficult to create a training dataset containing sufficient complete vehicles of different categories, especially trucks and buses. Therefore, fine-grained vehicle classification has only been realised via the clusters obtained from Objective (2). Vehicles are classified into different categories with F_1 score greater than 0.90 for all the four categories. F_1 score is larger than 0.95 for categories other than truck, which is higher than the reported highest classification accuracy 0.92 from a closely related work (Wu *et al.*, 2019).

6.2 Research contributions

The contributions of the research contained in this thesis can be summarised as:

1. Based on roadside lidar data, an integrated traffic monitoring system that can provide fundamental traffic information including the number of vehicles, vehicle dynamics, vehicle dimensions and vehicle types has been developed. The system has been demonstrated through different urban scenarios using two different lidar sensors to provide insight on real-world implementations of panoramic lidar sensors for traffic monitoring applications.

2. An advanced 3D object detection network, PV-RCNN, has been applied to our custom data to detect vehicles and pedestrians. The effectiveness of the network has been demonstrated by incremental improvement of the detection accuracy.

3. Two comprehensive vehicle tracking frameworks are presented in the traffic monitoring system to solve different problems. The first framework is intended to improve the accuracy of the estimated vehicle speeds through the use of a tracking refinement module. The true achievable accuracy of speed estimation using panoramic lidar was determined and the reliability of results are assured by independently validating the estimated speeds against an accurate vehicle speed reference system. The second framework aimed at improving the quality of the vehicle trajectories which might otherwise be shortened or interrupted in traditional strategy. In the meantime, the maximum tracking range of commonly used lidar sensors are determined by this framework, providing guidance for real-world lidar-based traffic applications.

4. A vehicle reconstruction module that is independent from any prior information or pre-trained deep learning models was proposed in this research. Reconstruction can be performed flexibly for any interested vehicles because it is based on vehicle trajectories with unique IDs. Moreover, two new applications of complete vehicle shapes can be prospected: 1) vehicle dimensions measurement; 2) fine-grained vehicle classification.

6.3 Summary

This research has developed a 3D lidar-based traffic monitoring system that can provide comprehensive traffic information through an end-to-end workflow, thereby determining

fundamental traffic parameters including the number of vehicles, vehicle dynamics, dimensions and types. Both traditional machine learning and deep learning methods have been experimented with vehicle detection, with the highest detection accuracy reaching 94%. A tracking refinement module has successfully improved the accuracy of determined vehicle speeds from 0.4m/s to 0.2m/s, surpassing the accuracy reported in contemporary literature. The quality of vehicle trajectories have been improved with regard to both the range and continuity by the joint vehicle detection and tracking framework. Complete vehicle shapes and vehicle types could also be obtained in addition to vehicle dynamics.

The results have demonstrated that roadside multi-beam lidar sensors are able to monitor traffic in various urban environments such as straight road sections, road intersections and crossroads. Main road users can be detected and tracked, where vehicles can be tracked at a range of speeds in line with typical urban environments. This work has also shown that a speed accuracy of c. 0.2m/s can be expected from high-precision 3D lidar data and this could benefit more detailed and accurate traffic flow or behaviour analysis. Trajectory interruption problems caused by light occlusion can be alleviated by detaching object tracking from object detection. The object tracking range of adopted lidar sensors can be extended to the furthest location where reflection from the object is extremely weak. It is concluded that the proposed system has the potential to be effectively employed for 3D urban traffic monitoring applications, but with the following limitations:

i) Heavy occlusion issues have not been successfully resolved where only a single lidar sensor is exploited.

ii) Real-time traffic monitoring cannot be realised in the current system as code is not optimized and computing power is limited, but near real-time processing can be achieved.

iii) The identification of vehicles is less straightforward than when using video imagery due to inherent lack of RGB information.

6.4 Future work

Based on the identified limitations of the developed system, the following aspects have been identified as potential areas to extend the work contained in the thesis:

- Multi-sensor utilization to address heavy occlusion issues

  In roadside laser scanning systems, vehicles in the furthest lanes from the sensor are likely to be occluded by those from the nearer lanes, especially in busy traffic. If the vehicle being tracked is completely occluded for some time, tracking as well as other related procedures would be affected. If there is another lidar sensor scanning from the other side of the road, this vehicle would be scanned. Sensors should be located properly to guarantee sufficient overlapped scanning region for point cloud registration. For the installation of the lidar sensors, refer to Section 5.2.4.

- Algorithm optimization to realize real-time traffic monitoring

  The proposed traffic monitoring system is not real-time, so optimization of the algorithms would be necessary in order to realize real-time traffic monitoring. Based on the real-time traffic monitoring system, urgent applications such as collision avoidance and anomaly detection can be enabled.

- Fusion of lidar sensor and video camera

  As pointed out in Chapter 5, one drawback of lidar data is the lack of textural information, which has made the identification of vehicles from lidar less straightforward. Therefore, database creation for vehicle reconstruction and vehicle classification in this research has to be aided by visually checking the video data. If lidar data and corresponding video data are fused, the above operations would become less laborious. From this aspect, fusing the raw data from lidar sensor and video camera is recommended in the future.

- Further exploration of PV-RCNN or other recently proposed networks to better facilitate 3D vehicle detection

  Two potential suggestions to better utilize PV-RCNN or PV-RCNN++ (Shi *et al.*, 2021) in the research would be to: i) adapt PV-RCNN(PV-RCNN++) from an object detector to an object

classifier. PV-RCNN was developed for object detection from the original lidar data. In this research, PV-RCNN was used on both the original lidar and the extracted moving points as a vehicle detector. Since moving points can be detected and object clusters subsequently extracted, 3D box proposals are generated. The remaining work is to classify these clusters into different categories using an object classifier. PV-RCNN can be used as the classifier after network adaption.

ii) apply PV-RCNN (PV-RCNN++) as a multi-class detector to directly obtain different categories of vehicles. In the current research, a hierarchical strategy was adopted: first vehicles were detected by PV-RCNN as one class and then fine-grained classified into different categories namely bus, car, truck and van. It is recommended in the future, research should regard PV-RCNN as a multi-class detector, which means bus, car, truck, van and pedestrian can be detected at the same time.

Utilization of latest object classifiers such as CurveNet (Xiang *et al.*, 2021) could also be a promising research aspect.

- Adoption of deep learning strategies for object tracking

  Deep learning strategies have very recently achieved state-of-the-art performance in object tracking in 3D points, while a traditional filtering approach was still employed for vehicle tracking in this research. For further improvement in the tracking performance, it is worth adopting an object tracking network into the vehicle tracking frameworks in the proposed traffic monitoring system. One example is a 'tracklet' proposal network named as PC-TCNN (Wu *et al*., 2021), for multi-object tracking on point clouds. This network first generates 'tracklet' proposals, then refines these 'tracklets' and associates them to generate long trajectories.

References

Advanced Navigation (2020) *Spatial Dual Manager* (Version 5.6) [Computer program]. Available at: https://www.advancednavigation.com/products/spatial-dual.

Advanced Navigation (2021) *Kinematica* [Computer program]. Available at: https://www.advancednavigation.com/products/kinematica.

Aijazi, A.K., Checchin, P., Malaterre, L. and Trassoudaine, L. (2016) 'Automatic detection of vehicles at road intersections using a compact 3D Velodyne sensor mounted on traffic signals', *2016 IEEE Intelligent Vehicles Symposium (IV)* IEEE, pp. 662-667.

Ali, S.S.M., George, B., Vanajakshi, L. and Venkatraman, J. (2011) 'A multiple inductive loop vehicle detection system for heterogeneous and lane-less traffic', *IEEE Transactions on Instrumentation and Measurement*, 61(5), pp. 1353-1360.

Ambardekar, A., Nicolescu, M., Bebis, G. and Nicolescu, M. (2014) 'Vehicle classification framework: a comparative study', *EURASIP Journal on Image and Video Processing*, 2014(1), pp. 1-13.

Arya Senna Abdul Rachman, A. (2017) '3D-LIDAR multi object tracking for autonomous driving: multi-target detection and tracking under urban road uncertainties'. Master thesis. Delft University of Technology. Available at: http://resolver.tudelft.nl/uuid:f536b829-42ae-41d5-968d-13bbaa4ec736 (Accessed: 14 September 2020).

Bakıcı, T., Almirall, E. and Wareham, J. (2013) 'A smart city initiative: the case of Barcelona', *Journal of the knowledge economy*, 4(2), pp. 135-148.

Bar‐Shalom, Y., Fortmann, T.E. and Cable, P.G. (1990) 'Tracking and data association'. *The journal of the Acoustical Society of America*, 87, pp. 918-919.

Baser, E., Balasubramanian, V., Bhattacharyya, P. and Czarnecki, K. (2019) 'Fantrack: 3d multi-object tracking with feature association network', *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1426-1433.

Beeferman, D. and Berger, A. (2000) 'Agglomerative clustering of a search engine query log', *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 407-416.

Bell, D., Xiao, W. and James, P. (2020) 'Accurate vehicle speed estimation from monocular camera footage', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 419-426.

Bencardino, M. and Greco, I. (2014) 'Smart communities. Social innovation at the service of the smart cities', *TeMA-Journal of Land Use, Mobility and Environment*.

Benevolo, C., Dameri, R.P. and D'auria, B. (2016) 'Smart mobility in smart city', in *Empowering organizations*. Springer, pp. 13-28.

Bergevin, R., Soucy, M., Gagnon, H. and Laurendeau, D. (1996) 'Towards a general multi-view registration technique', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5), pp. 540-547.

Besl, P.J. and McKay, N.D. (1992) 'Method for registration of 3-D shapes', *Sensor fusion IV: control paradigms and data structures*. International Society for Optics and Photonics, pp. 586-606.

Biber, P. and Straßer, W. (2003) 'The normal distributions transform: A new approach to laser scan matching', *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. IEEE, pp. 2743-2748.

Blackman, S.S. (2004) 'Multiple hypothesis tracking for multiple target tracking', *IEEE Aerospace and Electronic Systems Magazine*, 19(1), pp. 5-18.

Blais, G. and Levine, M.D. (1995) 'Registering multiview range data to create 3D computer objects', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), pp. 820-824.

Börcs, A. and Benedek, C. (2013) 'Urban traffic monitoring from lidar data with a two-level marked point process model'. Available at: https://eprints.sztaki.hu/id/eprint/7769 (Accessed: 23 November 2021).

Breiman, L. (2001) 'Random forests', *Machine learning*, 45(1), pp. 5-32.

Brenner, C. (2009) 'Extraction of features from mobile laser scanning data for future driver assistance systems', in *Advances in GIScience*. Springer, pp. 25-42.

Buch, N., Velastin, S.A. and Orwell, J. (2011) 'A review of computer vision techniques for the analysis of urban traffic', *IEEE Transactions on intelligent transportation systems*, 12(3), pp. 920-939.

Castellani, U., Fusiello, A. and Murino, V. (2002) 'Registration of multiple acoustic range views for underwater scene reconstruction', *Computer Vision and Image Understanding*, 87(1-3), pp. 78-89.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S. and Su, H. (2015) 'Shapenet: An information-rich 3d model repository', *arXiv preprint arXiv:1512.03012*.

Chen, Q.-A. and Tsukada, A. (2019) 'Detection-by-Tracking Boosted Online 3D Multi-Object Tracking', *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 295-301.

Chen, T., Vemuri, B.C., Rangarajan, A. and Eisenschenk, S.J. (2010) 'Group-wise point-set registration using a novel CDF-based Havrda-Charvát divergence', *International journal of computer vision*, 86(1), p. 111.

Chen, X., Ma, H., Wan, J., Li, B. and Xia, T. (2017) 'Multi-view 3d object detection network for autonomous driving', *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. pp. 1907-1915.

Chen, X., Wang, X. and Xuan, J. (2018) 'Tracking multiple moving objects using unscented Kalman filtering techniques', *arXiv preprint arXiv:1802.01235*.

Chen, Y. and Medioni, G. (1992) 'Object modelling by registration of multiple range images', *Image and vision computing*, 10(3), pp. 145-155.

Chetverikov, D., Svirko, D., Stepanov, D. and Krsek, P. (2002) 'The trimmed iterative closest point algorithm', *Object recognition supported by user interaction for service robots*. IEEE, pp. 545-548.

Cheung, S.-C.S. and Kamath, C. (2005) 'Robust background subtraction with foreground validation for urban traffic video', *EURASIP Journal on Advances in Signal Processing,* 2005(14), pp. 1-11.

Christodoulou, A. (2018) 'An image-based method for the pairwise registration of mobile laser scanning point clouds'. Master thesis. Delft University of Technology. Available at: http://resolver.tudelft.nl/uuid:5c1d2d90-8f2f-4dc2-b6c3-d5c635323c35 (Accessed: 10 September 2021).

Clark, D.E., Panta, K. and Vo, B.-N. (2006) 'The GM-PHD filter multiple target tracker', *2006 9th International Conference on Information Fusion*. IEEE, pp. 1-8.

Cohen, B. (2015) 'Urbanization, City growth, and the New United Nations development agenda', *Cornerstone*, 3(2), pp. 4-7.

Cui, Y., Xu, H., Wu, J., Sun, Y. and Zhao, J. (2019) 'Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system', *IEEE Intelligent Systems*, 34(3), pp. 44-51.

Dai, A., Ruizhongtai Qi, C. and Nießner, M. (2017) 'Shape completion using 3d-encoder-predictor cnns and shape synthesis', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5868-5877.

Dangi, V., Parab, A., Pawar, K. and Rathod, S.S. (2012) 'Image processing based intelligent traffic controller', Undergraduate Academic Research Journal (UARJ), 1(1), pp. 13-17.

De Feo, M., Graziano, A., Miglioli, R. and Farina, A. (1997) 'IMMJPDA versus MHT and Kalman filter with NN correlation: performance comparison', *IEE Proceedings-Radar, Sonar and Navigation*, 144(2), pp. 49-56.

Deep Systems (2017) *SUPERVISELY* [Computer program]. Available at: https://supervise.ly/.

Dieterle, T., Particke, F., Patino-Studencki, L. and Thielecke, J. (2017) 'Sensor data fusion of LIDAR with stereo RGB-D camera for object tracking', *2017 IEEE Sensors*. IEEE, pp. 1-3.

Doucet, A., Godsill, S. and Andrieu, C. (2000) 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Statistics and computing*, 10(3), pp. 197-208.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996) 'A density-based algorithm for discovering clusters in large spatial databases with noise', *kdd*. pp. 226-231.

Evangelidis, G.D. and Horaud, R. (2017) 'Joint alignment of multiple point sets with batch and incremental expectation-maximization', *IEEE transactions on pattern analysis and machine intelligence*, 40(6), pp. 1397-1410.

Evangelidis, G.D., Kounades-Bastian, D., Horaud, R. and Psarakis, E.Z. (2014) 'A generative model for the joint registration of multiple point sets', *European Conference on Computer Vision*. Springer, pp. 109-122.

Fraley, C. and Raftery, A.E. (1998) 'How many clusters? Which clustering method? Answers via model-based cluster analysis', *The computer journal*, 41(8), pp. 578-588.

Frank, L., Kavage, S. and Litman, T. (2006) 'Promoting public health through smart growth: Building healthier communities through transportation and land use policies and practices'. Available at: http://www.smartgrowth.bc.ca/downloads/SGBC_Health%20Report%20Final.pdf. (Accessed: 20 August 2021)

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013) 'Vision meets robotics: The kitti dataset', *The International Journal of Robotics Research*, 32(11), pp. 1231-1237.

Giraldo, L.G.S., Hasanbelliu, E., Rao, M. and Principe, J.C. (2017) 'Group-wise point-set registration based on Renyi's second order entropy', *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2454-2462.

Godin, G., Rioux, M. and Baribeau, R. (1994) 'Three-dimensional registration using range and intensity information', *Videometrics III*. International Society for Optics and Photonics, pp. 279-290.

Gonzalez, H., Rodriguez, S. and Elouardi, A. (2019) 'Track-Before-Detect Framework-Based Vehicle Monocular Vision Sensors', *Sensors*, 19(3), p. 560.

Graham, B., Engelcke, M. and Van Der Maaten, L. (2018) '3d semantic segmentation with submanifold sparse convolutional networks', *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 9224-9232.

Haritha, H. and Kumar, T.S. (2017) 'Survey on various traffic monitoring and reasoning techniques', *Computer Science On-line Conference*. Springer, pp. 507-516.

Hollands, R.G. (2008) 'Will the real smart city please stand up? Intelligent, progressive or entrepreneurial?', *City*, 12(3), pp. 303-320.

Hong, H. and Lee, B.H. (2017) 'Probabilistic normal distributions transform representation for accurate 3D point cloud registration', *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3333-3338.

Huang, K. and Hao, Q. (2021) 'Joint Multi-Object Detection and Tracking with Camera-LiDAR Fusion for Autonomous Driving', 2021 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6983-6989.

Hussain, K.F. and Moussa, G.S. (2005) 'Laser intensity vehicle classification system based on random neural network', *Proceedings of the 43rd annual Southeast regional conference-Volume 1*. pp. 31-35.

Im, H., Hong, B., Jeon, S. and Hong, J. (2016) 'Bigdata analytics on CCTV images for collecting traffic information', *2016 International Conference on Big Data and Smart Computing (BigComp)*. IEEE, pp. 525-528.

Iwasaki, Y., Misumi, M. and Nakamiya, T. (2013) 'Robust vehicle detection under various environmental conditions using an infrared thermal camera and its application to road traffic flow monitoring', *Sensors*, 13(6), pp. 7756-7773.

Jain, N.K., Saini, R. and Mittal, P. (2019) 'A review on traffic monitoring system techniques', in *Soft Computing: Theories and Applications*. Springer, pp. 569-577.

Jeong, T.T. (2007) 'Particle PHD filter multiple target tracking in sonar image', *IEEE Transactions on Aerospace and Electronic Systems*, 43(1), pp. 409-416.

Jokela, M., Kutila, M. and Pyykönen, P. (2019) 'Testing and validation of automotive point-cloud sensors in adverse weather conditions', *Applied Sciences*, 9(11), p. 2341.

Kalman, R.E. (1960) 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering*, 82(1), pp. 35-45.

Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. and Wu, A.Y. (2002) 'An efficient k-means clustering algorithm: Analysis and implementation', *IEEE transactions on pattern analysis and machine intelligence*, 24(7), pp. 881-892.

Kazhdan, M., Bolitho, M. and Hoppe, H. (2006) 'Poisson surface reconstruction', *Proceedings of the fourth Eurographics symposium on Geometry processing*.

Khan, M.A., Ectors, W., Bellemans, T., Janssens, D. and Wets, G. (2017) 'UAV-based traffic analysis: A universal guiding framework based on literature survey', *Transportation research procedia*, 22, pp. 541-550.

Khan, N.A., Jhanjhi, N., Brohi, S.N., Usmani, R.S.A. and Nayyar, A. (2020) 'Smart traffic monitoring system using unmanned aerial vehicles (UAVs)', *Computer Communications*, 157, pp. 434-443.

Ki, Y.-K. and Baik, D.-K. (2006a) 'Model for accurate speed measurement using double-loop detectors', *IEEE Transactions on Vehicular Technology*, 55(4), pp. 1094-1101.

Ki, Y.-K. and Baik, D.-K. (2006b) 'Vehicle-classification algorithm for single-loop detectors using neural networks', *IEEE Transactions on Vehicular Technology*, 55(6), pp. 1704-1711.

Ki, Y.-K., Choi, J.-W., Joun, H.-J., Ahn, G.-H. and Cho, K.-C. (2017) 'Real-time estimation of travel speed using urban traffic information system and cctv', *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, pp. 1-5.

Kim, C., Li, F., Ciptadi, A. and Rehg, J.M. (2015) 'Multiple hypothesis tracking revisited', *Proceedings of the IEEE international conference on computer vision*. pp. 4696-4704.

Klein, L.A., Mills, M.K., Gibson, D. and Klein, L.A. (2006) Traffic detector handbook: Volume II. United States. Federal Highway Administration. Available at:https://rosap.ntl.bts.gov/view/dot/936 (Accessed: 02 September 2020).

Kuhn, H.W. (1955) 'The Hungarian method for the assignment problem', *Naval research logistics quarterly*, 2(1‐2), pp. 83-97.

Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J. and Beijbom, O. (2019) 'Pointpillars: Fast encoders for object detection from point clouds', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 12697-12705.

Langerwisch, M. and Wagner, B. (2010) 'Registration of Indoor 3D Range Images using Virtual 2D Scans', *ICINCO (2)*. pp. 327-332.

Lazaroiu, G.C. and Roscia, M. (2012) 'Definition methodology for the smart cities model', *Energy*, 47(1), pp. 326-332.

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

Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D. and Mitra, N.J. (2011) 'Globfit: Consistently fitting primitives by discovering global relations', in *ACM SIGGRAPH 2011 papers*. pp. 1-12.

Liang, Y.-B., Qiu, Y. and Cui, T.-J. (2016) 'Semiautomatic registration of terrestrial laser scanning data using perspective intensity images', *IEEE Geoscience and Remote Sensing Letters*, 14(1), pp. 28-32.

Lin, C.-C., Tai, Y.-C., Lee, J.-J. and Chen, Y.-S. (2017) 'A novel point cloud registration using 2D image features', *EURASIP Journal on Advances in Signal Processing*, 2017(1), pp. 1-11.

Lin, W.-H., Dahlgren, J. and Huo, H. (2004) 'Enhancement of vehicle speed estimation with single loop detectors', *Transportation research record*, 1870(1), pp. 147-152.

Lin, Y.-L., Morariu, V.I., Hsu, W. and Davis, L.S. (2014) 'Jointly optimizing 3d model fitting and fine-grained classification', *European conference on computer vision*. Springer, pp. 466-480.

Liu, S., Luo, J., Hu, J., Luo, H. and Liang, Y. (2021) 'Research on NDT-based Positioning for Autonomous Driving', *E3S Web of Conferences*. EDP Sciences, pp. 02055.

Liu, Y., Kong, D., Zhao, D., Gong, X. and Han, G. (2018) 'A Point Cloud Registration Algorithm Based on Feature Extraction and Matching', *Mathematical Problems in Engineering*, 2018, pp. 1-9.

Liu, Y., Tian, B., Chen, S., Zhu, F. and Wang, K. (2013) 'A survey of vision-based vehicle detection and tracking techniques in ITS', *Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, pp. 72-77.

Luo, Z., Habibi, S. and Mohrenschildt, M.v. (2016) 'LiDAR based real time multiple vehicle detection and tracking', *International Journal of Computer and Information Engineering*, 10(6), pp. 1125-1132.

Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S., Li, X., Si, Z. and Guo, J. (2019) 'Fine-grained vehicle classification with channel max pooling modified CNNs', *IEEE Transactions on Vehicular Technology*, 68(4), pp. 3224-3233.

Mahler, R.P. (2003) 'Multitarget Bayes filtering via first-order multitarget moments', *IEEE Transactions on Aerospace and Electronic systems*, 39(4), pp. 1152-1178.

Mathews, A.A. and Babu, A. (2017) 'Automatic Number Plate Detection', *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, pp. 50-55.

Meta, S. and Cinsdikici, M.G. (2010) 'Vehicle-classification algorithm based on component analysis for single-loop inductive detector', *IEEE Transactions on Vehicular Technology*, 59(6), pp. 2795-2805.

Milan, A., Rezatofighi, S.H., Dick, A., Reid, I. and Schindler, K. (2017) 'Online multi-target tracking using recurrent neural networks', *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence,* pp. 4225-4232.

Mitzel, D. and Leibe, B. (2012) 'Taking Mobile Multi-object Tracking to the Next Level: People, Unknown Objects, and Carried Items', *European Conference on Computer Vision*. Springer, pp. 566-579.

Morris, B.T., Tran, C., Scora, G., Trivedi, M.M. and Barth, M.J. (2012) 'Real-time video-based traffic measurement and visualization system for energy/emissions', *IEEE Transactions on Intelligent Transportation Systems*, 13(4), pp. 1667-1678.

Nguyen, D.T., Hua, B.-S., Tran, K., Pham, Q.-H. and Yeung, S.-K. (2016) 'A field model for repairing 3d shapes', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5676-5684.

Nüchter, A., Lingemann, K., Hertzberg, J. and Surmann, H. (2005) 'Heuristic-based laser scan matching for outdoor 6D SLAM', *Annual Conference on Artificial Intelligence*. Springer, pp. 304-319.

Ordnance Survey (2016) *Grid InQuestII* [Computer program]. Available at: https://www.ordnancesurvey.co.uk/business-government/tools-support/osnet/transformation.

Ošep, A., Mehner, W., Voigtlaender, P. and Leibe, B. (2018) 'Track, then decide: Category-agnostic vision-based multi-object tracking', *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1-8.

Padilla, R., Netto, S.L. and da Silva, E.A. (2020) 'A survey on performance metrics for object-detection algorithms', *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, pp. 237-242.

Park, M.-W., Kim, J.I., Lee, Y.-J., Park, J. and Suh, W. (2017) 'Vision-based surveillance system for monitoring traffic conditions', *Multimedia Tools and Applications*, 76(23), pp. 25343-25367.

Pauly, M., Mitra, N.J., Giesen, J., Gross, M.H. and Guibas, L.J. (2005) 'Example-based 3d scan completion', *Symposium on Geometry Processing*. pp. 23-32.

Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H. and Guibas, L.J. (2008) 'Discovering structural regularity in 3D geometry', in *ACM SIGGRAPH 2008 papers*. pp. 1-11.

Pham, N.T., Huang, W. and Ong, S.H. (2007) 'Tracking multiple objects using probability hypothesis density filter and color measurements', *2007 IEEE International Conference on Multimedia and Expo*. IEEE, pp. 1511-1514.

Philipp Glira (2015a) *General description of the globalICP method*. Available at: https://www.geo.tuwien.ac.at/downloads/pg/pctools/globalICPGeneralDescription.html (Accessed: 15 June 2020).

Philipp Glira (2015b) *Point cloud tools for Matlab*. Available at: https://www.geo.tuwien.ac.at/downloads/pg/pctools/pctools.html#ICP (Accessed: 15 June 2020).

Pinto, J.A., Kumar, P., Alonso, M.F., Andreão, W.L., Pedruzzi, R., dos Santos, F.S., Moreira, D.M. and de Almeida Albuquerque, T.T. (2020) 'Traffic data in air quality modelling: a review of key variables, improvements in results, open problems and challenges in current research', *Atmospheric Pollution Research*, 11(3), pp. 454-468.

Pomerleau, F., Colas, F., Siegwart, R. and Magnenat, S. (2013) 'Comparing ICP variants on real-world data sets', *Autonomous Robots*, 34(3), pp. 133-148.

Pöschmann, J., Pfeifer, T. and Protzel, P. (2020) 'Factor graph based 3d multi-object tracking in point clouds', *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 10343-10350.

Puri, A. (2005) *A survey of unmanned aerial vehicles (UAV) for traffic surveillance*. *Department of computer science and engineering*, University of South Florida, pp.1-29.

Pyykönen, P., Molinier, M. and Klunder, G. (2010) 'Traffic monitoring and modelling for intersection safety', *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*. IEEE, pp. 401-408.

Qi, C.R., Litany, O., He, K. and Guibas, L.J. (2019) 'Deep hough voting for 3d object detection in point clouds', *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9277-9286.

Qi, C.R., Liu, W., Wu, C., Su, H. and Guibas, L.J. (2018) 'Frustum pointnets for 3d object detection from rgb-d data', *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 918-927.

Qi, C.R., Su, H., Mo, K. and Guibas, L.J. (2017a) 'Pointnet: Deep learning on point sets for 3d classification and segmentation', *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 652-660.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017b) 'Pointnet++: Deep hierarchical feature learning on point sets in a metric space', *31st Conference on Neural Information Processing Systems*, pp. 1-10.

RAC (2017) *Traffic cameras - what you need to know*. Available at: https://www.rac.co.uk/drive/advice/cameras/traffic-cameras/.

Radosavljević, Z. (2006) 'A study of a target tracking method using Global Nearest Neighbor algorithm', *Vojnotehnički glasnik*, 54(2), pp. 160-167.

Ravikumar, N., Gooya, A., Çimen, S., Frangi, A.F. and Taylor, Z.A. (2018) 'Group-wise similarity registration of point sets using Student's t-mixture model for statistical shape models', *Medical image analysis*, 44, pp.156-176.

Reid, D. (1979) 'An algorithm for tracking multiple targets', *IEEE transactions on Automatic Control*, 24(6), pp. 843-854.

Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A. and Reid, I. (2015) 'Joint probabilistic data association revisited', *Proceedings of the IEEE international conference on computer vision*. pp. 3047-3055.

Robert, K. (2009) 'Video-based traffic monitoring at day and night vehicle features detection tracking', *2009 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 1-6.

Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D. and Hoiem, D. (2015) 'Completing 3d object shape from one depth image', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2484-2493.

Rousseeuw, P.J. (1984) 'Least median of squares regression', *Journal of the American statistical association*, 79(388), pp. 871-880.

Sanchez, A., Suárez, P.D., Conci, A. and Nunes, E. (2010) 'Video-based distance traffic analysis: Application to vehicle tracking and counting', *Computing in Science & Engineering*, 13(3), pp. 38-45.

Sanchez, J., Denis, F., Checchin, P., Dupont, F. and Trassoudaine, L. (2017) 'Global registration of 3D LiDAR point clouds based on scene features: Application to structured environments', *Remote Sensing*, 9(10), p. 1014.

Saunier, N. and Sayed, T. (2006) 'A feature-based tracking algorithm for vehicles in intersections', *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE, pp. 59-59.

Schnabel, R., Degener, P. and Klein, R. (2009) 'Completion and reconstruction with primitive shapes', *Computer Graphics Forum*, 28(2), pp. 503-512.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (2017) 'DBSCAN revisited, revisited: why and how you should (still) use DBSCAN', *ACM Transactions on Database Systems*, 42(3), pp. 1-21.

Serna, A. and Marcotegui, B. (2014) 'Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning', *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, pp. 243-255.

Sharma, A., Grau, O. and Fritz, M. (2016) 'Vconv-dae: Deep volumetric shape learning without object labels', *European Conference on Computer Vision*. Springer, pp. 236-250.

Sharp, G.C., Lee, S.W. and Wehe, D.K. (2002) 'ICP registration using invariant features', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), pp. 90-102.

Shen, C.-H., Fu, H., Chen, K. and Hu, S.-M. (2012) 'Structure recovery by part assembly', *ACM Transactions on Graphics (TOG)*, 31(6), pp. 1-11.

Shetty, A.P. (2017) *GPS-LiDAR sensor fusion aided by 3D city models for UAVs*. Master thesis. University of Illinois at Urbana-Champaign. Available at: http://hdl.handle.net/2142/97501 (Accessed: 10 September 2021).

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X. and Li, H. (2020a) 'Pv-rcnn: Point-voxel feature set abstraction for 3d object detection', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529-10538.

Shi, S., Guo, C., Yang, J. and Li, H. (2020b) 'PV-RCNN: The Top-Performing LiDAR-only Solutions for 3D Detection/3D Tracking/Domain Adaptation of Waymo Open Dataset Challenges', *arXiv preprint arXiv:2008.12599*.

Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X. and Li, H. (2021) 'PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection', *arXiv preprint arXiv:2102.00463*.

Shi, S., Wang, X. and Li, H. (2019a) 'Pointrcnn: 3d object proposal generation and detection from point cloud', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770-779.

Shi, S., Wang, Z., Shi, J., Wang, X. and Li, H. (2020c) 'From points to parts: 3d object detection *from point cloud with part-aware and part-aggregation network', IEEE transactions on pattern analysis and machine intelligence, 43(8), pp. 2647-2664.*

Shi, X., Peng, J., Li, J., Yan, P. and Gong, H. (2019b) 'The iterative closest point registration algorithm based on the normal distribution transformation', *Procedia Computer Science*, 147, pp. 181-190.

Shirazi, M.S. and Morris, B.T. (2016) 'Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies', *IEEE Transactions on Intelligent Transportation Systems*, 18(1), pp. 4-24.

Slama, M.A.Y. (2017) 'Normal Distribution Transform with Point Projection for 3D Point Cloud Registration'. *5th International Conference on Control & Signal Processing*, pp. 117-121.

Song, S. and Xiao, J. (2016) 'Deep sliding shapes for amodal 3d object detection in rgb-d images', *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 808-816.

Song, X., Liu, S., Guo, S., Wu, J., Li, Z., Zhang, H. and Pi, R. (2021) 'Road users classification technology based on roadside light detection and ranging', *International Conference on Sensors and Instruments 2021*. SPIE.

Staricco, L. (2013) 'Smart Mobility Opportunities and Conditions', *TeMA-Journal of Land Use, Mobility and Environment*, 6(3), pp. 342-354.

Stark, M., Krause, J., Pepik, B., Meger, D., Little, J.J., Schiele, B. and Koller, D. (2011) 'Fine-grained categorization for 3d scene understanding', *International Journal of Robotics Research*, 30(13), pp. 1543-1552.

Sualeh, M. and Kim, G.-W. (2019) 'Dynamic multi-lidar based multiple object detection and tracking', *Sensors*, 19(6), p. 1474.

Sulaiman, H.A.B., Afif, M., Othman, M.A.B., Misran, M.H.B. and Said, M. (2013) 'Wireless based smart parking system using zigbee', *IJET*, 5, pp. 3282-3300.

Sung, M., Kim, V.G., Angst, R. and Guibas, L. (2015) 'Data-driven structural priors for shape completion', *ACM Transactions on Graphics (TOG)*, 34(6), pp. 1-11.

Trucco, E., Fusiello, A. and Roberto, V. (1999) 'Robust motion and correspondence of noisy 3-D point sets with missing data', *Pattern recognition letters*, 20(9), pp. 889-898.

Syarif, I., Prugel-Bennett, A. and Wills, G. (2016) 'SVM parameter optimization using grid search and genetic algorithm to improve classification performance', *Telkomnika*, 14(4), p. 1502.

Taek Lee, J. and Chung, Y. (2017) 'Deep learning-based vehicle classification using an ensemble of local expert and global networks', *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 47-52.

Tagliasacchi, A., Olson, M., Zhang, H., Hamarneh, G. and Cohen-Or, D. (2011) 'Vase: Volume-aware surface evolution for surface reconstruction from incomplete point clouds', *Computer Graphics Forum*, 30(5), pp. 1563-1571.

Tian, B., Yao, Q., Gu, Y., Wang, K. and Li, Y. (2011) 'Video processing techniques for traffic flow monitoring: A survey', *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1103-1108.

Tong, H., Zhang, H., Meng, H. and Wang, X. (2010) 'Multitarget Tracking Before Detection via Probability Hypothesis Density Filter', *2010 International Conference on Electrical and Control Engineering*. IEEE, pp. 1332-1335.

Trucco, E., Fusiello, A. and Roberto, V. (1999) 'Robust motion and correspondence of noisy 3-D point sets with missing data', *Pattern recognition letters*, 20(9), pp. 889-898.

United Nations (2014) *World Urbanization Prospects*. Available at: https://digitallibrary.un.org/record/3833745?ln=en (Accessed: 20 September 2021)

Varley, J., DeChant, C., Richardson, A., Ruales, J. and Allen, P. (2017) 'Shape completion enabled robotic grasping', *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 2442-2447.

Velodyne. (2016). VLP-16 manual: User's manual and programming guide. Velodyne LiDAR. Inc. https://usermanual.wiki/Pdf/VLP1620User20Manual20and20Programming20Guide206392432 0Rev20A.1947942715/view

Vo, B.-N. and Ma, W.-K. (2006) 'The Gaussian mixture probability hypothesis density filter', *IEEE Transactions on signal processing*, 54(11), pp. 4091-4104.

Wang, D., Huang, C., Wang, Y., Deng, Y. and Li, H. (2020a) 'A 3D Multiobject Tracking Algorithm of Point Cloud Based on Deep Learning', *Mathematical Problems in Engineering*, 2020, pp. 1-10.

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.

Wang, S., Sun, Y., Liu, C. and Liu, M. (2020b) 'PointTrackNet: An End-to-End Network For 3-D Object Detection and Tracking From Point Clouds', *IEEE Robotics and Automation Letters*, 5(2), pp. 3206-3212.

Weng, X. and Kitani, K. (2019a) 'A baseline for 3d multi-object tracking', *arXiv preprint arXiv:1907.03961*, 1(2), p.6.

Weng, X. and Kitani, K. (2019b) 'Monocular 3d object detection with pseudo-lidar point cloud', *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. pp. 1-14.

Weng, X., Wang, J., Held, D. and Kitani, K. (2020a) '3d multi-object tracking: A baseline and new evaluation metrics', *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 10359-10366.

Weng, X., Wang, Y., Man, Y. and Kitani, K.M. (2020b) 'Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 6499-6508.

Wojtanowski, J., Zygmunt, M., Kaszczuk, M., Mierczyk, Z. and Muzal, M. (2014) 'Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions', *Opto-Electronics Review*, 22(3), pp. 183-190.

Wu, J. (2018a) 'An automatic procedure for vehicle tracking with a roadside LiDAR sensor', *Institute of Transportation Engineers. ITE Journal*, 88(11), pp. 32-37.

Wu, J. (2018b) *Data processing algorithms and applications of LiDAR-enhanced connected infrastructure sensing*. PhD thesis. University of Nevada. Available at: http://hdl.handle.net/11714/4871 (Accessed: 21 November 2021)

Wu, J., Xu, H., Sun, Y., Zheng, J. and Yue, R. (2018) 'Automatic Background Filtering Method for Roadside LiDAR Data', *Transportation Research Record*, 2672(45), pp. 106-114.

Wu, J., Xu, H., Tian, Y., Pi, R. and Yue, R. (2020) 'Vehicle detection under adverse weather from roadside LiDAR data', *Sensors*, 20(12), p. 3433.

Wu, J., Xu, H., Zheng, Y., Zhang, Y., Lv, B. and Tian, Z. (2019) 'Automatic vehicle classification using roadside LiDAR data', *Transportation Research Record*, 2673(6), pp. 153-164.

Wu, J., Zhang, Y., Tian, Y., Yue, R. and Zhang, H. 'Automatic Vehicle Tracking with LiDAR-Enhanced Roadside Infrastructure', *Journal of Testing and Evaluation*, 49(1), pp.121-133.

Wu, S., Huang, H., Gong, M., Zwicker, M. and Cohen-Or, D. (2015a) 'Deep points consolidation', *ACM Transactions on Graphics (ToG)*, 34(6), pp. 1-13.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. (2015b) '3d shapenets: A deep representation for volumetric shapes', *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912-1920.

Wu, H., Li, Q., Wen, C., Li, X., Fan, X. and Wang, C. (2021) ' Tracklet Proposal Network for Multi-Object Tracking on Point Clouds', *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1165-1171.

Xia, Y., Liu, W., Luo, Z., Xu, Y. and Stilla, U. (2020a) 'COMPLETION OF SPARSE AND PARTIAL POINT CLOUDS OF VEHICLES USING A NOVEL END-TO-END NETWORK', *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, pp.933-940.

Xia, Y., Xu, Y., Wang, C. and Stilla, U. (2020b) 'VPC-Net: Completion of 3D Vehicles from MLS Point Clouds', *ISPRS Journal of Photogrammetry and Remote Sensing*, 174, pp.166-181.

Xiang, T., Zhang, C., Song, Y., Yu, J. and Cai, W. (2021) 'Walk in the Cloud: Learning Curves for Point Clouds Shape Analysis', *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 915-924.

Xiao, W., Vallet, B., Brédif, M. and Paparoditis, N. (2015) 'Street environment change detection from mobile laser scanning point clouds', *ISPRS Journal of Photogrammetry and Remote Sensing*, 107, pp. 38-49.

Xiao, W., Vallet, B., Schindler, K. and Paparoditis, N. (2016a) 'Simultaneous Detection and Tracking of Pedestrian from Velodyne Laser Scanning Data', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, pp. 295-302.

Xiao, W., Vallet, B., Schindler, K. and Paparoditis, N. (2016b) 'Street-side vehicle detection, classification and change detection using mobile laser scanning data', *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, pp. 166-178.

Xiao, W., Vallet, B., Xiao, Y., Mills, J. and Paparoditis, N. (2017) 'OCCUPANCY MODELLING FOR MOVING OBJECT DETECTION FROM LIDAR POINT CLOUDS: A COMPARATIVE STUDY', *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, pp. 171-178.

Xu, H., Tian, Z., Wu, J., Liu, H. and Zhao, J. (2018) *High-Resolution Micro Traffic Data From Roadside LiDAR Sensors for Connected-Vehicles and New Traffic Applications*. University of Nevada, Reno. Solaris University Transportation Centre.

Xue, S., Li, G., Lv, Q., Meng, X. and Tu, X. (2019) 'Point Cloud Registration Method for Pipeline Workpieces Based On NDT and Improved ICP Algorithms', *IOP Conference Series: Materials Science and Engineering*. IOP Publishing, p. 022131.

Yan, Y., Mao, Y. and Li, B. (2018) 'Second: Sparsely embedded convolutional detection', *Sensors*, 18(10), p. 3337.

Yang, J., Wang, C., Luo, W., Zhang, Y., Chang, B. and Wu, M. (2021) 'Research on Point Cloud Registering Method of Tunneling Roadway Based on 3D NDT-ICP Algorithm', *Sensors*, 21(13), p. 4448.

Yao, W., Hinz, S. and Stilla, U. (2011) 'Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis', *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), pp. 260-271.

Yao, W., Zhang, M., Hinz, S. and Stilla, U. (2012) 'Airborne traffic monitoring in large areas using LiDAR data–theory and experiments', *International journal of remote sensing*, 33(12), pp. 3930-3945.

Yu, S., Wu, Y., Li, W., Song, Z. and Zeng, W. (2017) 'A model for fine-grained vehicle classification based on deep learning', *Neurocomputing*, 257, pp. 97-103.

Zhang, J., Pi, R., Ma, X., Wu, J., Li, H. and Yang, Z. (2021) 'Object Classification with Roadside LiDAR Data Using a Probabilistic Neural Network', *Electronics*, 10(7), p. 803.

Zhao, J. (2019) *Exploring the fundamentals of using infrastructure-based LiDAR sensors to develop connected intersections*. PhD thesis. Texas Tech University. Available at: https://hdl.handle.net/2346/85580 (Accessed: 21 November 2021).

Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y. and Wu, D. (2019) 'Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors', *Transportation research part C: emerging technologies*, 100, pp. 68-87.

Zhang, J., Xiao, W., Coifman, B. and Mills, J.P. (2020) 'Vehicle Tracking and Speed Estimation From Roadside Lidar', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, pp. 5597-5608.

Zhang, Z., Zheng, J., Xu, H., Wang, X., Fan, X. and Chen, R. (2019) 'Automatic background construction and object detection based on roadside LiDAR', *IEEE Transactions on Intelligent Transportation Systems*, 21(10), pp. 4086-4097.

Zhang, L., Wang, F., Hu, M., Shi, L. and Liang, L. (2013) 'A vehicle counting algorithm using foreground detection in traffic videos', *Proceedings of the 3rd Interna-tional Conference on Multimedia Technology. Paris: Atlantis Press*. Citeseer, pp. 232-239.

Zhao, J., Xu, H., Tian, Y. and Liu, H. (2020) 'Towards application of light detection and ranging sensor to traffic detection: an investigation of its built-in features and installation techniques', Journal of Intelligent Transportation Systems, 26(2), pp. 213-234.

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N.J., Cohen-Or, D. and Chen, B. (2010) 'Non-local scan consolidation for 3D urban scenes', *ACM Trans. Graph.*, 29(4), pp. 94:1-94:9.

Zhou, Y. and Tuzel, O. (2018) 'Voxelnet: End-to-end learning for point cloud based 3d object detection', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4490-4499.

Zhu, H., Guo, B., Zou, K., Li, Y., Yuen, K.-V., Mihaylova, L. and Leung, H. (2019b) 'A review of point set registration: From pairwise registration to groupwise registration', *Sensors*, 19(5), p. 1191.

Zia, M.Z., Stark, M. and Schindler, K. (2015) 'Towards scene understanding with detailed 3d object representations', *International Journal of Computer Vision*, 112(2), pp. 188-203.