

2017

Multi-Sensor Data Fusion for Robust Environment Reconstruction in Autonomous Vehicle Applications

Abu Hasnat Mohammad Rubaiyat

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Rubaiyat, Abu Hasnat Mohammad, "Multi-Sensor Data Fusion for Robust Environment Reconstruction in Autonomous Vehicle Applications" (2017). *Graduate Theses, Dissertations, and Problem Reports*. 6537. <https://researchrepository.wvu.edu/etd/6537>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

MULTI-SENSOR DATA FUSION FOR ROBUST ENVIRONMENT RECONSTRUCTION IN AUTONOMOUS VEHICLE APPLICATIONS

Abu Hasnat Mohammad Rubaiyat

**Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University**

in partial fulfillment of the requirements for the degree of

**Master of Science in
Electrical Engineering**

Yaser P. Fallah, Ph.D., Chair

Xin Li, Ph.D.

Parviz Famouri, Ph.D.

Muhammad A. Choudhry, Ph.D.

**Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia**

2017

Keywords: multi-sensor data fusion; irregularities; vehicle environment; V2V; DSRC

Copyright 2017 Abu Hasnat Mohammad Rubaiyat

ABSTRACT

MULTI-SENSOR DATA FUSION FOR ROBUST ENVIRONMENT RECONSTRUCTION IN AUTONOMOUS VEHICLE APPLICATIONS

Abu Hasnat Mohammad Rubaiyat

In autonomous vehicle systems, understanding the surrounding environment is mandatory for an intelligent vehicle to make every decision of movement on the road. Knowledge about the neighboring environment enables the vehicle to detect moving objects, especially irregular events such as jaywalking, sudden lane change of the vehicle etc. to avoid collision. This local situation awareness mostly depends on the advanced sensors (e.g. camera, LIDAR, RADAR) added to the vehicle. The main focus of this work is to formulate a problem of reconstructing the vehicle environment using point cloud data from the LIDAR and RGB color images from the camera. Based on a widely used point cloud registration tool such as iterated closest point (ICP), an expectation-maximization (EM)-ICP technique has been proposed to automatically mosaic multiple point cloud sets into a larger one. Motion trajectories of the moving objects are analyzed to address the issue of irregularity detection. Another contribution of this work is the utilization of fusion of color information (from RGB color images captured by the camera) with the three-dimensional point cloud data for better representation of the environment. For better understanding of the surrounding environment, histogram of oriented gradient (HOG) based techniques are exploited to detect pedestrians and vehicles.

Using both camera and LIDAR, an autonomous vehicle can gather information and reconstruct the map of the surrounding environment up to a certain distance. Capability of communicating and cooperating among vehicles can improve the automated driving decisions by providing extended and more precise view of the surroundings. In this work, a transmission power control algorithm is studied along with the adaptive content control algorithm to achieve a more accurate map of the vehicle environment. To exchange the local sensor data among the vehicles, an adaptive communication scheme is proposed that controls the lengths and the contents of the messages depending on the load of the communication channel. The exchange of this information can extend the tracking region of a vehicle beyond the area sensed by its own sensors. In this experiment, a combined effect of power control, and message length and content control algorithm is exploited to improve the map's accuracy of the surroundings in a cooperative automated vehicle system.

ACKNOWLEDGEMENTS

It is a pleasure to express my sincere gratitude to my advisor, Dr. Yaser P. Fallah, for his supervision throughout the course of my research. His continuous encouragement and suggestions helped me to boost my research skills.

I would also like to convey my heartfelt thanks to Dr. Xin Li for his continuous guidance. His advice and innovative ideas helped me to successfully complete my thesis. I greatly appreciate my thesis committee members, Dr. Parviz Famouri and Dr. Muhammad A. Choudhry, for their time and support. I wish to acknowledge the Lane Department of Computer Science and Electrical Engineering to give me the opportunity of pursuing my master's degree in such an academically vibrant environment.

I would like to thank my family for supporting me throughout my studies. I am also thankful to the members of my research group for their help during my research at WVU.

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1 Literature Survey	3
1.2 Contributions	6
1.3 Organization of the Thesis	7
Chapter 2: Environment Reconstruction	8
2.1 Analysis of Multi-Sensor Data	8
2.2 Reconstruct Scenario from Multiple Point Clouds	9
2.2.1 Alignment of Multiple Point Clouds: Hierarchical Extension of ICP	10
2.2.2 Distinguish Between Moving and Steady Objects	12
2.3 Multi-Sensor Data Fusion	17
2.4 Irregularity Detection	19
2.4.1 Tracking Moving Objects	20
2.4.2 Predict Irregularities using Linear Regression	21
2.5 Vehicle Environment Understanding	24
2.5.1 Pedestrian Detection from Point Cloud	25
2.5.2 Vehicle Detection using PIHOG	30
Chapter 3: Power Control in V2V	37
3.1 V2V Communications System	37
3.1.1 Dedicated Short Range Communication (DSRC)	37
3.2 System Model and Communication Architecture	38
3.3 Message Length and Content Control Scheme	41
3.4 Proposed Power Control Scheme	42
3.4.1 Effect of Transmission Power on Map Accuracy	43
3.4.2 Power Control Strategy	49
Chapter 4: Experimental Results and Discussions	51
4.1 Environment Reconstruction and Understanding	51
4.1.1 EM-ICP based Environment Reconstruction	51
4.1.2 Motion Trajectory Analysis and Irregularity Detection	53
4.1.3 Pedestrian Detection Results	55
4.1.4 Vehicle Detection Results	56

4.2	Performance of Power Control Algorithm	59
Chapter 5:	Conclusion and Future Works.....	61

LIST OF FIGURES

Figure 2-1: 3D point cloud representation of a scenario	8
Figure 2-2: RGB image of the scenario shown in Figure 2-1	9
Figure 2-3: Distribution of matching errors in ICP algorithm.....	12
Figure 2-4: Point cloud representation of an urban road (a) before and (b) after alignment	13
Figure 2-5: Bird view of a road in residential area (a) before and (b) after removing the road surface	15
Figure 2-6: Detected moving points between two consecutive point cloud frames	16
Figure 2-7: A moving cyclist in multiple frames of a point cloud data sequence	17
Figure 2-8: 6D model of the scenario shown in Figure 2-1	19
Figure 2-9: Tracking multiple moving objects from LIDAR data.....	21
Figure 2-10: Predicted irregular movement of a cyclist (red color object)	24
Figure 2-11: Clustering result- (a) original point cloud, (b) clusters are shown in different colors and numbers.....	27
Figure 2-12: 2D d-maps of (a) pedestrian, and (b) non-human clusters	27
Figure 2-13: Detected pedestrian (red colored) in the scenario shown in Figure 2-11.....	30
Figure 2-14: Two different cells with same histogram of gradients [47].....	31
Figure 2-15: 2D front view of the 6D map shown in Figure 2-8.....	32
Figure 2-16: (a) Mean, and (b) standard deviation all positive vehicle images	35
Figure 2-17: Detected vehicles (in red boxes) using PIHOG-SVM	36
Figure 3-1: Cooperative automated vehicle system model	39
Figure 3-2: Road topology used in the simulation [22]	40
Figure 3-3: Scalable communication architecture proposed in [27].....	41
Figure 3-4: PTE vs distance for (a)125 vehicles/Km, and (b) 250 vehicles/Km.....	44
Figure 3-5: Information age vs distance for (a) 125 vehicles/Km, and (b) 250 vehicles/Km.....	45
Figure 3-6: CBR vs time for (a) 125 vehicles/Km, and (b) 250 vehicles/Km.....	46
Figure 3-7: Message length vs time for (a) 125 vehicles/Km, and (b) 250 vehicles/Km	47
Figure 3-8: PER vs distance for (a) 125 vehicles/Km, (b) 250 vehicles/Km	48
Figure 4-1: Reconstructed vehicle environments from a collection of 10 point clouds- a) urban vs. b) rural.....	52
Figure 4-2: Reconstructed vehicle environments from a collection of 10 point clouds- a) road vs. b) intersection	52
Figure 4-3: Alignment accuracy comparison (ground-truth is supplied by KITTI [54]).....	53
Figure 4-4: Irregularity detection in different scenario (a) pedestrian, (b) cyclist, (c) vehicle	54
Figure 4-5: Results of HOG-SVM based pedestrian detection	55
Figure 4-6: Results of PIHOG-SVM based vehicle detection.....	58
Figure 4-7: PTE comparison among three schemes for vehicle densities (a) 125 vehicles/Km, and (b) 250 vehicles/Km.....	60

LIST OF TABLES

Table 3-1: Parameters used in simulation to detect optimum	50
Table 4-1: Overall and frame-wise pedestrian detection accuracy	56
Table 4-2: Overall and frame-wise vehicle detection accuracy.....	58

LIST OF ACRONYMS

Acronym	Definition
AVS	Autonomous Vehicle System
RADAR	Radio Detection and Ranging
LIDAR	Light Detection and Ranging
FPS	Frame per Second
RMSE	Root Mean Square Error
ICP	Iterative Closest Point
EM	Expectation-Maximization
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
HOG	Histogram of Oriented Gradient
PIHOG/ π HOG	Position and Intensity – included HOG
SVM	Support Vector Machine
DSRC	Dedicated Short-Range Communications
V2V	Vehicle to Vehicle
PTE	Position Tracking Error
CBR	Channel Busy Ratio
PER	Packet Error Rate
VANET	Vehicular Ad hoc Network

Chapter 1: Introduction

Each year tens of thousands of drivers and passengers die in United States because of road accidents; most of the fatal crashes involve more than one vehicle. For many years, researchers are trying to develop a fully autonomous system of transportation to reduce on-road fatalities and save lives. An autonomous vehicle system (AVS) is a combination of some automated systems where perception, decision making, and operation of the automobile are performed by electronics and machinery instead of a human driver. This system includes the control of vehicle movements, destination, path planning, awareness of vehicle environment, intervehicle communication, emergency awareness etc. Successful reconstruction of neighboring environment is a must for safety awareness. Automated vehicles use a variety of sensing modalities for the environment reconstruction, including radar, lidar, and camera. As the imaging technology has immensely progressed in recent years, cameras are widely used in AVS. Small size, low price, and availability of cameras are also other reasons behind this. Use of LIDAR point cloud data is the relatively newer addition to the AV system. The main working principle of measuring the distance to a target using LIDAR is to illuminate the target with a pulsed laser light, and then measure the reflected pulses with a LIDAR sensor (e.g. Velodyne HDL-16e, 32e, 64e). Differences in laser return times are used to make digital 3D-representations of the target, which contains XYZ information of the points on that target. Analysis of LIDAR data in AVS has gained popularity because LIDAR admits computationally more efficient processing and enjoys less vulnerability to adversary weather and lighting conditions comparing with the image data. Many researchers have used data from these two sensors both separately and combinedly to gather the environment knowledge. In this work, we have discussed an approach to vehicle

environment reconstruction and understanding based on the fusion of data from these two local sensors (i.e. camera and LIDAR).

Having the knowledge about the surrounding environment is essential in the autonomous vehicle system (AVS). Environment related knowledge can be used in the safety applications (e.g. collision avoidance, emergency warning) and efficient route planning. Various local sensors, such as cameras, LIDAR, or RADAR are mainly used in automated vehicle applications to gather information of its surroundings and make necessary driving decisions. But, relying only on these sensors can refrain the vehicle from getting complete view of its neighborhood, as these sensors have limitations in terms of range and occlusion. One solution to mitigate this issue and expand the reach of local information available at each vehicle is the use of short range wireless communication among the vehicles. Local sensors in a vehicle can gather its speed, location and destination information, and generate a map containing vehicles, pedestrians, road-side infrastructure and other objects in its neighborhood. In cooperative vehicle system, vehicles communicate the locally generated maps to each other and thus expand their maps.

Local sensors can supply a rich source of data that can be exploited by intelligent vehicles for reconstructing the surrounding environment, however, the cost and accuracy of different sensors (e.g., camera vs. LIDAR) would vary [1] [2]. Different image based roadside object detection (e.g. pedestrian, vehicle) techniques have been studied in many literatures [3] [4] [5] [6] [7]. Other than images, LIDAR-based data (point cloud) has received increasingly more attention in recent years due to the advance in both hardware (e.g., from Velodyne HDL-16e, 32e to 64e) and software (i.e., point cloud registration [8] [9] [10], segmentation [11] and classification [12]). When compared against image data, LIDAR admits computationally more efficient processing and enjoys less vulnerability to adversary weather and lighting conditions. Under the context of intelligent vehicles,

LIDAR-based approaches have been studied for the detection/tracking of pedestrian [13] [14] curb [15] [16] and vehicle [17] [12]. Combination of RGB image and point cloud data has also been studied in some articles [18] [19].

In the networked vehicular application, dedicated short-range communication (DSRC) is playing the leading role as a key technology to enable communication among the vehicles [20]. But the systems that depend on the broadcast of information over DSRC networks most of the time suffer from scalability challenges [21]. In case of high channel loads, DSRC throughput degrades due to high vehicle density, high transmission rate, or large packet size. Some scalable channel load control strategies based on the message rate and transmission power control are extensively studied in the literature [22] [23] [24] [21] [25] [26]. An adaptive, probabilistic and distance dependent strategy was proposed in [27], which controls the length and contents of the message exchanged among vehicles based on the communication channel load. But, it can be shown that even for optimum message size, performance can degrade if the vehicle does not maintain proper transmission power.

1.1 Literature Survey

The whole experiment can be divided into two parts: local sensors based environment reconstruction and understanding, and proposing a power control algorithm for DSRC based V2V communications. Many researchers have worked on both issues. A similar problem to the reconstruction of surface from point cloud data has been studied in [28]. This is a classical problem that has been extensively studied by computer graphics community. However, in intelligent vehicle system, efficient reconstruction and visualization of point cloud data often faces new challenges due to their large size and arising from the desire of exploiting structural a priori information (e.g., urban vs. rural environment). Because of the moving objects (including the host vehicle) the vehicle environment

has the tendency to keep changing. Therefore, registration of multiple point clouds plays a vital role in vehicle environment reconstruction from 3D data. The idea of the 3D data registration was developed in 1990s based on the seminal work of iterative closest point (ICP) [29]. Since then, many variants of ICP techniques have been developed and applied to various data sets (e.g., [9], [10], [30]). Most of these existing works have shown good performance on clean and complete datasets. The issue of outlier detection or robustness has not been addressed in the open literature as far as we know.

To understand the reconstructed environment, different point cloud clustering and segmentation algorithms have been studied in the literatures. Like image data, clustering and segmentation of point cloud data can be handy in supporting other high-level tasks such as classification and recognition. For clustering and segmentation most of the existing works (e.g., [11], [31], [12], [32]) focused on the use of the geometric features/clues. When the objective is to separate moving objects from still background (vehicle environment) for scene analysis especially for the purpose of detecting irregular events, we argue it is more effective to exploit motion-related information (e.g., trajectory, speed and acceleration).

Several techniques have been proposed to detect pedestrian using both images and 3D point cloud. Dalal and Triggs [33] proposed a human detection scheme that uses SIFT-inspired [34] features, called histograms of oriented gradients (HOGs), and a linear SVM as a learning method. This technique has widely been used by the researchers in pedestrian detection problems. Papageorgiou and Poggio [35] exploited the Haar wavelets (HWs) as features to train a quadratic support vector machines (SVMs) with front- and rear-viewed pedestrians. Viola and Jones [36], [37] proposed AdaBoost cascades (layers of threshold-rule weak classifiers) as a learning algorithm to exploit Haar-like features for surveillance-oriented pedestrian detection. In [38] and [39], Gavrilu et al. used

image gray-scale pixels as features and a neural network with local receptive fields (NN-LRFs [40]) as the learning machine. Zhao and Thorpe used image gradient magnitude and a feedforward neural network [41]. All these articles mentioned above used images to extract features. Pedestrian detection using 3D LIDAR data has recently received attention [42], [43]. Hegger et al. used Local Surface Normals in 3D point cloud for people detection [44]. A layered approach to find pedestrian in 3D data has been proposed by Spinello et al. [45]. The main idea of this approach is to subdivide a pedestrian into parts defined by different height levels, and learn a highly specialized classifier for each part.

Like the pedestrian, vehicle detection techniques are also applied to both images and 3D point cloud [46]. Position and intensity-include HOG based vehicle detection technique has been proposed in [47]. Erbs et al. analyzed stixel dynamics to detect moving vehicles from images [48]. Vehicle detection from 3D point cloud data has also been discussed in several articles. Li et al. used fully convolutional network to detect vehicle from 3D LIDAR data [49]. Eum et al. presented a novel method for vehicle detection from airborne LiDAR point clouds based on a decision tree algorithm with horizontal and vertical features of the segment [50]. In our experiment, we have used fusion based technique (using both camera and LIDAR) for environment understanding. Similar approaches have been discussed in the articles [18], [19] for pedestrian and vehicle detection respectively. We have proposed to generate a 6D representation of the neighborhood using fusion of camera and LIDAR data to understand the environment.

The second part of this experiment is to propose a transmission power control algorithm combined with an adaptive content control algorithm to achieve a more accurate and extended map of the neighborhood. DSRC based V2V communication system has been used to maintain the communication among the intelligent vehicles. In case of high channel loads the systems that depend

on the broadcast of information over DSRC networks most of the time suffer from scalability challenges [21], i.e. DSRC throughput degrades due to high vehicle density, high transmission rate, or large packet size. Some scalable channel load control strategies based on the message rate and transmission power control are extensively studied in the literature [21] [22] [23] [24] [25] [51] [52]. An adaptive, probabilistic and distance dependent strategy was proposed in [27], which controls the length and contents of the message exchanged among vehicles based on the communication channel load. In our experiment, we have adopted this message length and content control scheme. A power control algorithm has been coupled with this scheme to achieve an optimum message length that can represent the environment map more accurately.

1.2 Contributions

The main contributions of this experiment can be stated as follows:

1. A robust point cloud mosaicking technique has been proposed combining iterative closest point (ICP) based registration [29] and EM-based background estimation [53].
2. Fusion of data from multiple sensors (i.e. camera and LIDAR) has been utilized to reconstruct the vehicle environment.
3. An irregularity detection technique has been proposed based on the motion trajectories of the moving objects (e.g. walking pedestrians, moving vehicles).
4. Histogram of oriented gradient (HOG) based object detection techniques have been used to detect pedestrians and vehicles under the context of vehicle environment understanding.
5. A combination of message content control and adaptive power control algorithm is proposed to achieve high level of accuracy in generation of the extended neighborhood map.

1.3 Organization of the Thesis

The whole methodology of this experiment can be organized in two different parts. The first part describes the multi-sensor data fusion based vehicle environment reconstruction and understanding (Chapter 2:), and the next part explains the power control algorithm (coupled with adaptive content control algorithm) for DSRC vehicle-to-vehicle (V2V) communications system (Chapter 3:). Chapter 2: illustrates the environment reconstruction technique using RGB image and point cloud, irregularity detection based on trajectories of moving objects, and HOG based pedestrian and vehicle detection techniques. The proposed environment reconstruction and understanding techniques mainly use the data from two of the local sensors, camera and LIDAR. LIDAR gives the XYZ coordinates of the points on the objects, and camera provides the color information of that points. We have used fusion of the data from these two local sensors to reconstruct the map of vehicle environment. Like other traditional sensors, these two also have limitations. Objects beyond the reach of the sensors cannot be located using the local sensors. To see beyond the range of the sensors vehicle-to-vehicle (V2V) communication is used where intelligent vehicles share their map information (gathered using local sensors) with each other, and thus update and extend their own maps. The accuracy of this map varies with different factors, e.g. transmission power, transmission frequency, channel load, message length etc. Chapter 3: discusses about the effects of varying transmission power in DSRC based intervehicle communication system, and proposes a power control scheme combined with an adaptive message length and content control algorithm to improve the map accuracy of the intelligent vehicle system. Result section is included in Chapter 4: . Finally, we conclude and mention some future works in Chapter 5: .

Chapter 2: Environment Reconstruction

2.1 Analysis of Multi-Sensor Data

To understand the neighborhood, an autonomous vehicle uses different sensors like camera, RADAR, LIDAR etc. In this work, data from camera and LIDAR are analyzed to reconstruct the vehicle environment. Using camera RGB images of the roads and other objects are captured. LIDAR sensor continuously collects a collection of point cloud data surrounding the host vehicle. Each of those point cloud data carry partial information about the environment and therefore it is desirable to develop a method that can collectively combine multiple point clouds into one whole set. From point cloud data, XYZ position of a point on an object can be derived. But for the color information we need the RGB image of that object. So, a technique that can combine point cloud data and color information, can provide a better representation of the neighboring environment of the host vehicle than using the sensor information separately. Figure 2-1 shows a point cloud representation of a sample scenario (from KITTI database [54]) in three dimensions. The RGB image of the same scenario has been illustrated in Figure 2-2.

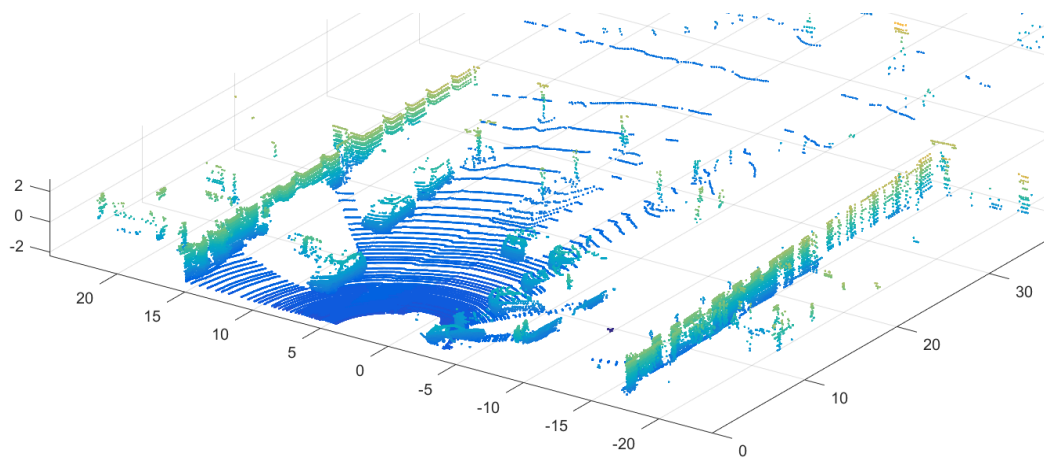


Figure 2-1: 3D point cloud representation of a scenario



Figure 2-2: RGB image of the scenario shown in Figure 2-1

2.2 Reconstruct Scenario from Multiple Point Clouds

In KITTI dataset, the point cloud data are collected using a Velodyne laser scanner (LIDAR) which is mounted on a moving autonomous vehicle. The LIDAR sensor continuously gathers point cloud data of the surrounding environment at 10fps frame rate. Each frame contains partial information of the environment. So, we need to come up with an alignment technique to combine these frames into one to get an overall image of the surroundings. This environment reconstruction problem is analogous to the image mosaicking problem. Unlike image mosaicking, in case of 3D point clouds it is often difficult to find salient feature points like corners or SIFT key points to facilitate the alignment. The only exception is a Point Feature Histograms (PFH) based technique described in [31], but high complexity is a big issue here. In this experiment, we have exploited iterative closest point (ICP) [29] based technique to align multiple point cloud simultaneously.

2.2.1 Alignment of Multiple Point Clouds: Hierarchical

Extension of ICP

Iterative closest point (ICP) [29] and its variants [1] [9] [10] are some of the widely-used tools to align two point clouds. To align multiple ($N > 2$) point clouds $\{P_1, P_2, \dots, P_N\}$, an extension of ICP is required. One ad-hoc approach is to align first two point clouds and then sequentially group the rest of the point clouds with it, e.g. P_1 and P_2 into $P_{1,2}$, then group $P_{1,2}$ and P_3 into $P_{1,2,3}$ and so on. In other words, first point cloud P_1 is used as the reference and the remaining $N-1$ point clouds are aligned with respect to P_1 . Problem may arise if N is large or the speed of the vehicle is high; e.g. due to the motion of self-driving vehicle, P_1 and P_N could have little overlap and therefore P_1 is not an appropriate reference while aligning P_N . If the constant speed model is assumed for the vehicle, the optimal reference frame derived for aligning both P_1 and P_N , should be the frame in the middle. Applying this argument recursively to the first and second halves of N point clouds, we can obtain the following hierarchical extension of ICP algorithm:

- Perform pairwise alignment to N point clouds and obtain $N/2$ point clouds after merging; i.e. $\{P_{1,2}, P_{3,4}, \dots, P_{N-1,N}\}$
- Same process is performed on the merged point clouds (newly created) recursively until only one is left

The described hierarchical strategy of extending ICP nicely fits octree-based 3D data representation, e.g. Octomap [55]. The main advantage of tree-based representation is it offers great flexibility of attaining the tradeoff between complexity and accuracy.

Speed of the host vehicle has some effects in the performance of the alignment process. As the speed of the host vehicle decreases to zero, the overlap among the adjacent point cloud frames (P_i vs. P_{i+1})

would increase which implies more computations (as more frames would be processed) are required for environment reconstruction. To reduce the wastage of computation, one can easily skip pairwise alignment when the speed of the vehicle is very low, e.g. instead of processing P_1 and P_2 , we can use P_1 and P_3 or P_4 in case of very low vehicle speed (equivalent to temporarily down-sampling the point cloud dataset). As the vehicle starts to move faster, less overlap occurs, which can increase the probability of misalignment. So, finding the optimal vehicle speed for environment reconstruction is necessary to achieve better alignment result.

In this experiment, error generated by ICP algorithm has been taken into consideration to predict the vehicle speed. The basic idea is to find the optimal transformation (translation and rotation) such that the matching errors between two consecutive point cloud frames can be minimized. Root mean square error (RMSE) is calculated in each iteration of ICP algorithm. In first iteration, zero movement of the host vehicle is assumed in between two consecutive point cloud frames. Vehicle movement is increased by 0.2m in after iteration. The movement that gives the lowest RMSE is considered as the optimal vehicle movement between two frames. Speed of the vehicle can also be calculated from this information using velodyne frame rate. Figure 2-3 shows the distribution of matching errors (RMSE) in ICP algorithm as a function of vehicle displacement for the alignment of two consecutive point cloud frames. Minimum RMSE has been found for the displacement value 0.5m. So, the optimum movement of the vehicle between these two frames is 0.5m.

Figure 2-4 illustrates a result of the described alignment technique. Here, we applied the extended ICP based alignment process in an urban road for 10 frames. Figure 2-4(a) shows the combined point clouds for 10 frames before alignment, the result after alignment has been shown in (b).

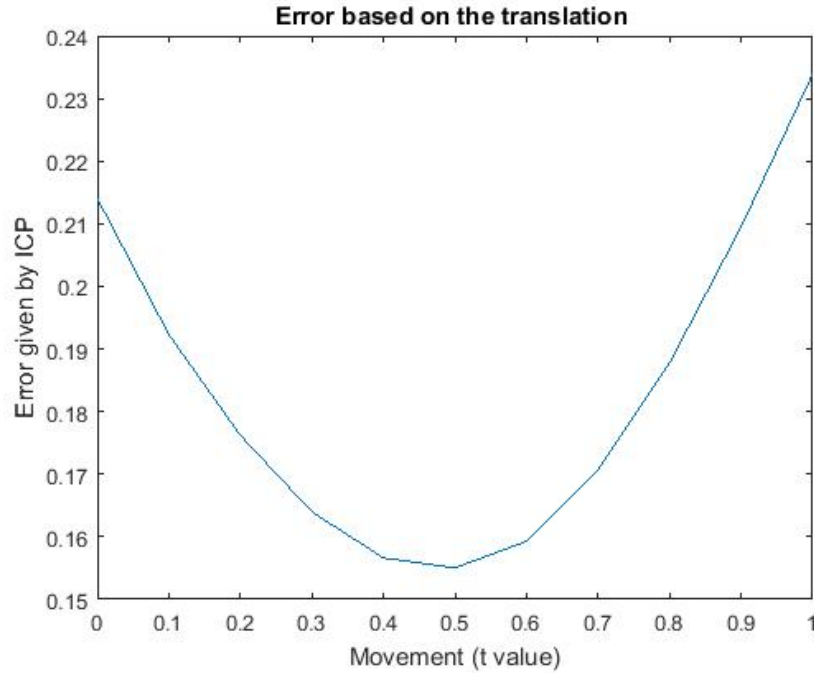
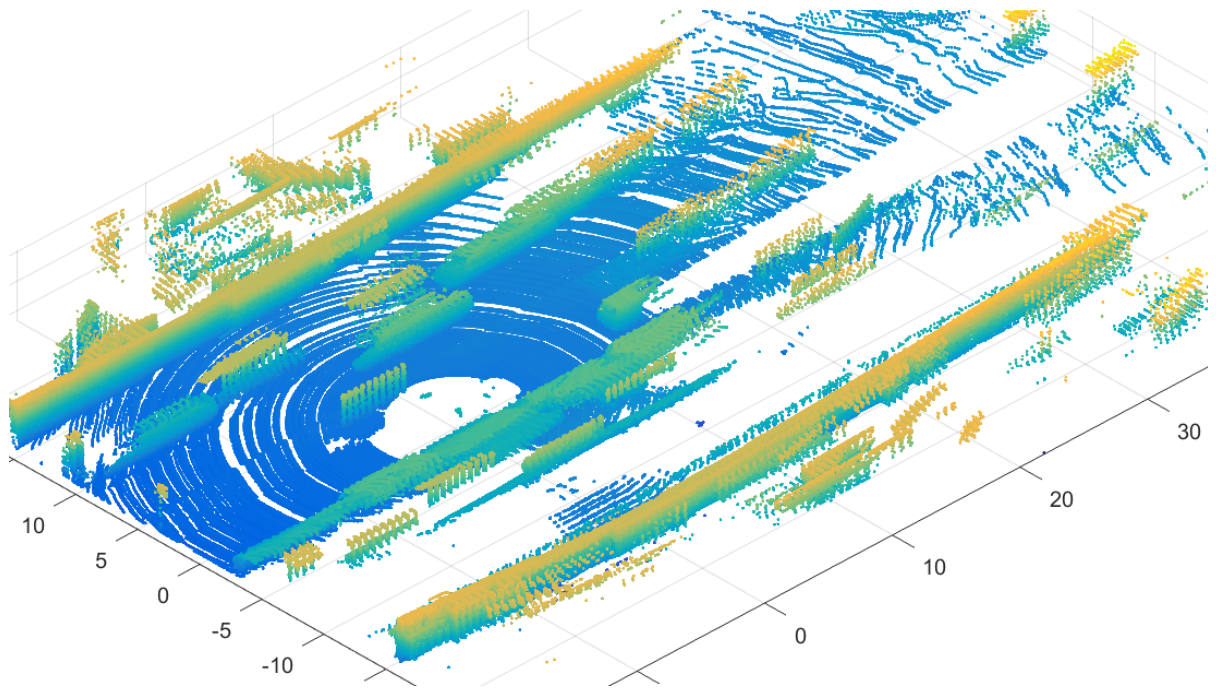


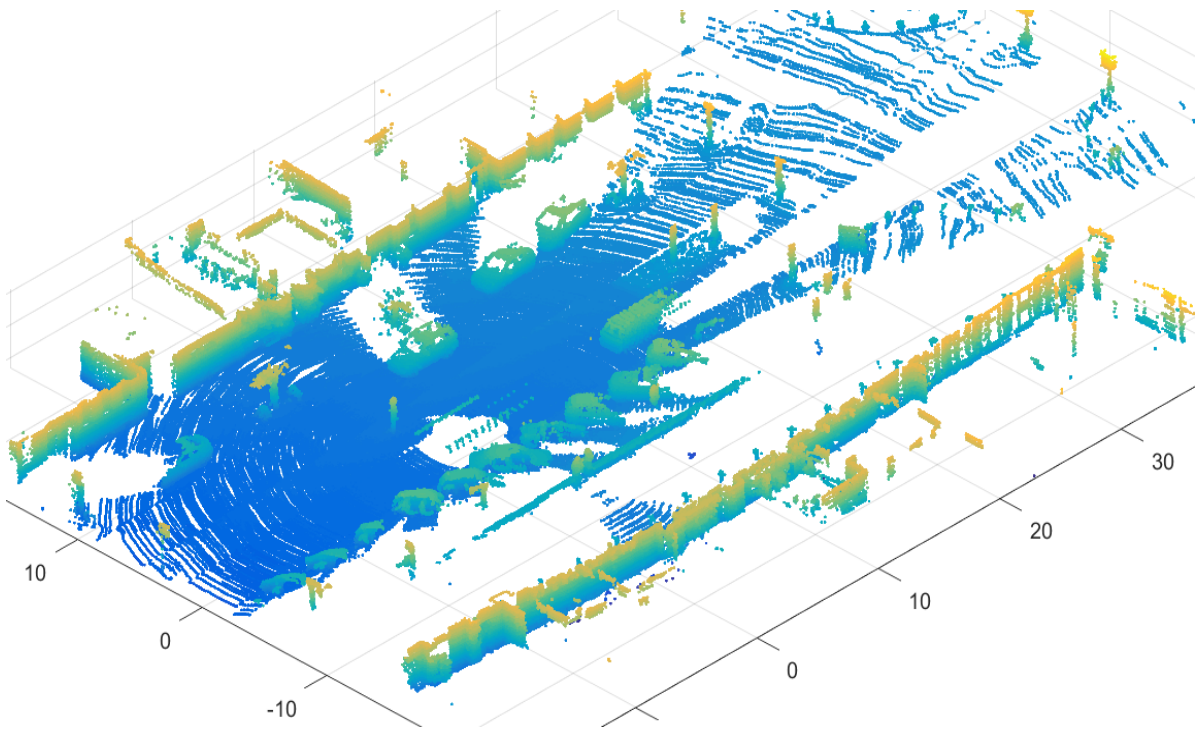
Figure 2-3: Distribution of matching errors in ICP algorithm

2.2.2 Distinguish Between Moving and Steady Objects

The resulting point cloud found after aligning a set of observed point clouds, can be considered as the reconstructed environment of the autonomous vehicle. It can be separated in two parts: the steady objects (e.g. road, trees, road-side buildings, pole, parked vehicles etc.) as background and moving objects (e.g. moving vehicles, motorcycles, walking pedestrians, cyclists etc.) as foreground. Under the context of autonomous vehicle applications, these two parts must be distinguished properly to understand the surrounding environment. Like the image mosaicking, moving objects can also create the interference (outlier-moving points) in point cloud mosaicking. Inspired by previous work on robust surface registration [10], a robust outlier (i.e. moving points) detection technique has been proposed in this work. Later moving objects are detected by applying a clustering algorithm to the moving points.



(a)



(b)

Figure 2-4: Point cloud representation of an urban road (a) before and (b) after alignment

2.2.2.1 Remove Points from Ground Level

To detect moving objects LIDAR data from multiple frames are processed. It is understandable that the points on the road surface can be included in the background (steady object). Hence, it is unnecessary to apply the moving object detection algorithm to these points. So, in the first step unnecessary points from ground level (i.e. road surface) are removed from the LIDAR data frame. To be precise, if one uses the Z-values of the point cloud (i.e., the altitude of a point) as the feature, road surface can be removed by thresholding the Z-values. That leaves the data points of all the objects (both steady and moving) on the road and the roadside environment. Figure 2-5(b) shows the result of the ground level removal technique applied to the point cloud shown in (a).

2.2.2.2 Moving Points Detection using Robust EM-ICP

As mentioned earlier, in point cloud registration or mosaicking interference from the moving objects are considered as the outliers. Hence, we have primarily considered the moving points as the outliers and applied a robust ICP based alignment technique with outlier rejection to detect those points. The main idea is based on the observation that point cloud registration and outlier detection have characteristics of chicken-and-egg problem – i.e. solving one problem creates path to the solution to the other. In this experiment, classical expectation-maximization (EM) method has been exploited into ICP as follows (similar approach is proposed in [56]),

- E-step: For an estimated transformation, align two point clouds and calculate the closest distance for each point. Declare a point to be an outlier if the closest distance is above a pre-selected threshold.
- M-step: For a given set of inliers, apply ICP to update the estimate of transformation parameters (i.e., translation and rotation).

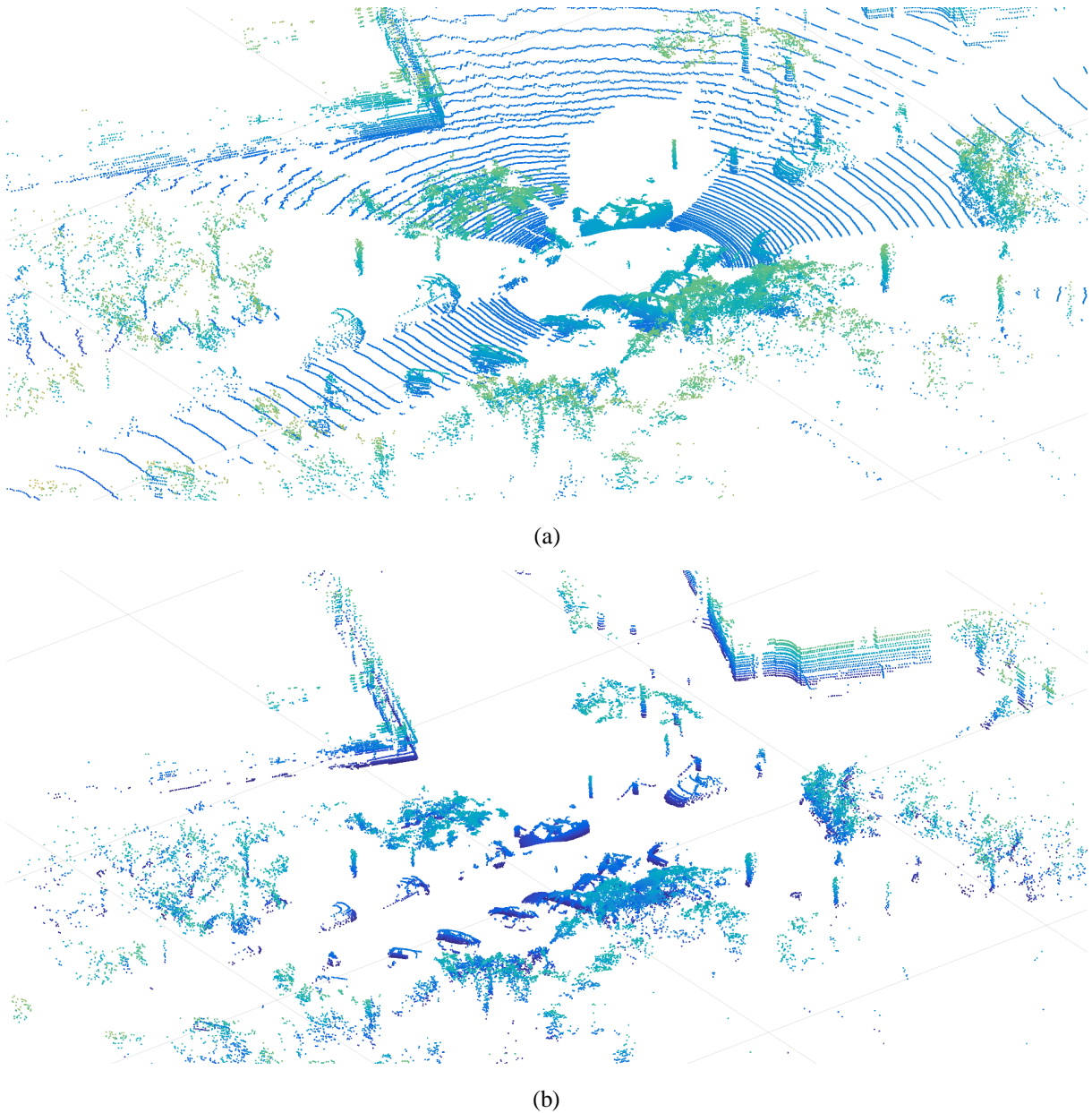


Figure 2-5: Bird view of a road in residential area (a) before and (b) after removing the road surface

The previous works of EM-ICP [10] [56] were applied to 3D point cloud sets of relatively small size ($\sim 10^4$). In this experiment, we are dealing with dozens of point clouds each containing around 10^6 samples. So, one of the contributions of this work is that this is at least the first experimental work of

testing the feasibility of EM-ICP on a large 3D dataset, e.g. KITTI dataset [54]. After executing this step, we get the moving points between two consecutive frames. Not all the moving points are part of actual moving objects. Some points are mistakenly detected and these points are observed in more scattered locations than the actual moving points (Figure 2-6). These outliers can be detected by exploiting any density-based clustering algorithm. Density-based spatial clustering of applications with noise (DBSCAN) [57] has been used to remove these outliers. DBSCAN is a density based clustering algorithm widely used in clustering problem for point clouds. It marks points as in same cluster that are closely packed together (nearby neighbors) and labels the points as outliers which are found in the low-density areas (nearest neighbors are far away). DBSCAN has been applied to the moving points found using the EM-ICP technique. Points that are packed in the same cluster, are considered to be from same object. So, number of moving objects is equal to the number of clusters. The clusters with less or equal than five points are considered as outliers (not moving object) and removed from the foreground. The remaining clusters are the actual moving objects that constitute the foreground of the reconstructed environment.

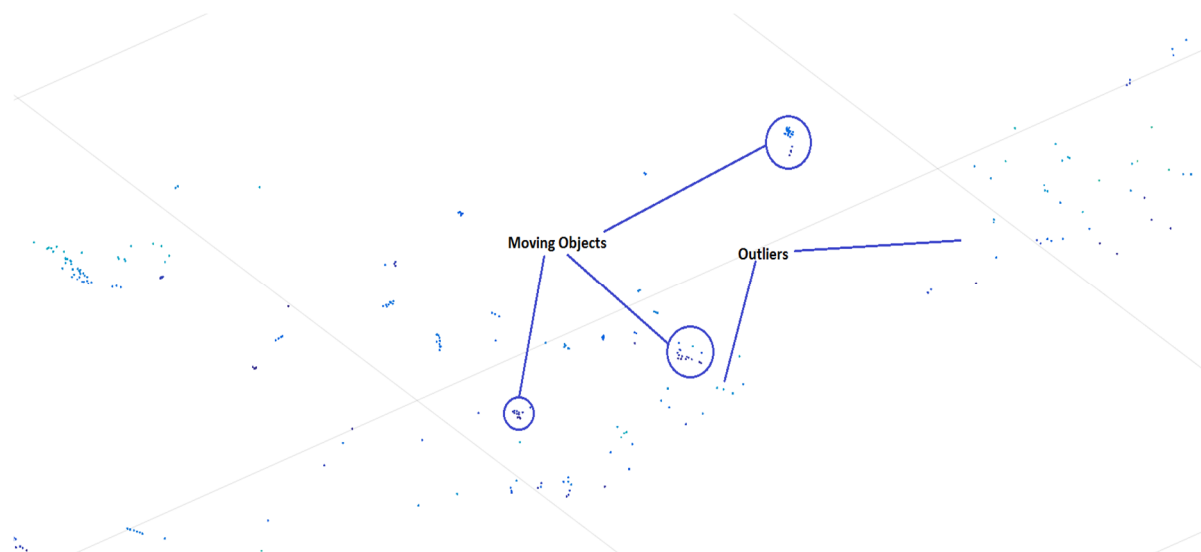


Figure 2-6: Detected moving points between two consecutive point cloud frames

Figure 2-7 shows the result of moving object detection technique. Here, a moving cyclist has been detected for 4 consecutive frames (total 5 frames are needed, first frame is for reference).

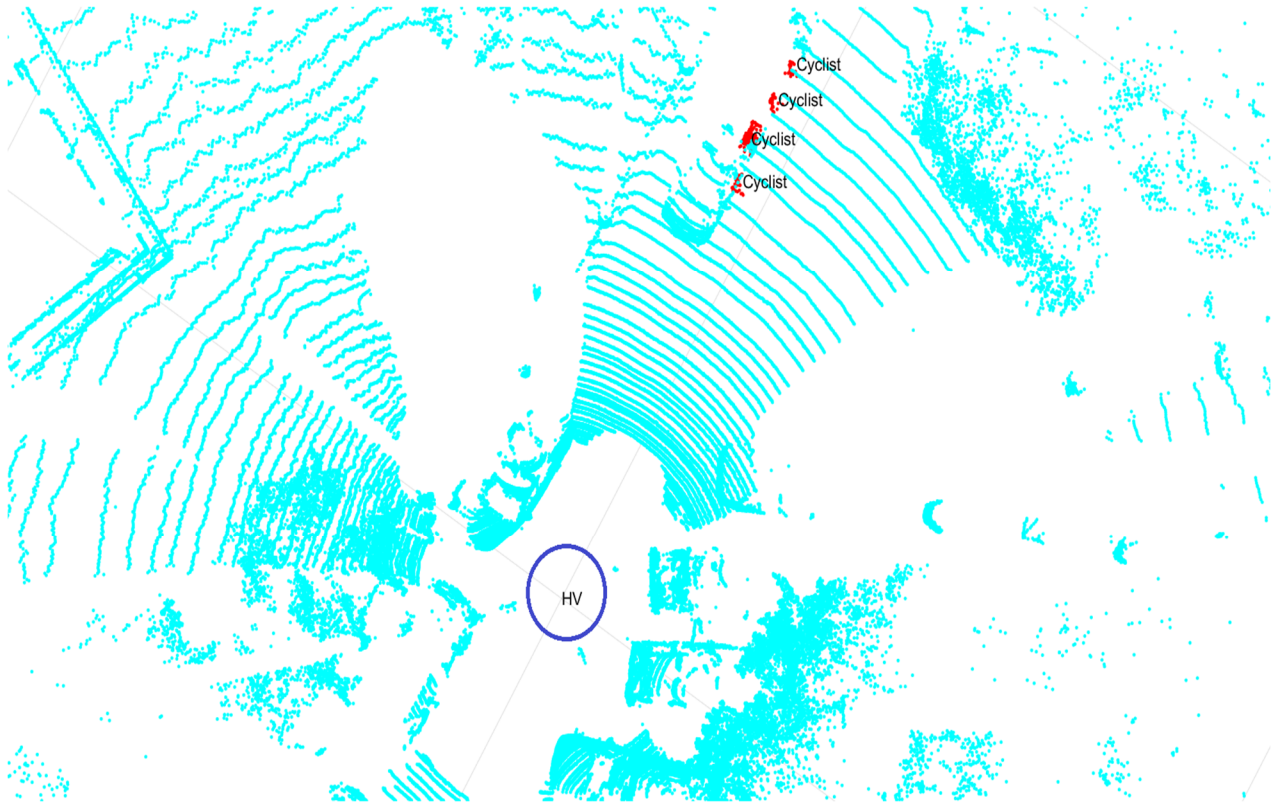


Figure 2-7: A moving cyclist in multiple frames of a point cloud data sequence

2.3 Multi-Sensor Data Fusion

In the previous sections of this chapter, we have discussed about the reconstruction of the vehicle environment using only LIDAR data. This gives us the 3D coordinates of different points from different points (both background and foreground). In this section, RGB color information from a camera has been combined with the coordinate information of the points to generate a six-dimensional model of the environment. First three dimensions are XYZ values of the point and last three dimensions contain RGB values of that point gathered from the color image.

The calibration of the velodyne coordinates with the camera has been done following the procedure described in the article [54]. A 3D point x in velodyne coordinates is projected to a point y in the image of the i^{th} camera using,

$$y = P_{rect}^i R_{rect}^i T_{velo}^{cam} x; P_{rect}^i \in \mathcal{R}^{3 \times 4}, R_{rect}^i \in \mathcal{R}^{3 \times 3} \dots \dots \dots (2-1)$$

Where, P_{rect}^i and R_{rect}^i are the projection matrix after rectification and rectifying rotation matrix of the camera respectively. Both parameters are used to calibrate the i^{th} camera. Here, $i \in \{0, 1, 2, 3\}$ is the camera index; 0 is for left grayscale, 1 is the right grayscale, 2 left color, and 3 right color camera. In this experiment, only left color camera ($i=2$) has been used. T_{velo}^{cam} is the velodyne to camera transformation matrix which is calculated using,

$$T_{velo}^{cam} = \begin{pmatrix} R_{velo}^{cam} & t_{velo}^{cam} \\ 0 & 1 \end{pmatrix} \dots \dots \dots (2-2)$$

- $R_{velo}^{cam} \in \mathcal{R}^{3 \times 3}$ is the rotation matrix: velodyne \rightarrow camera
- $t_{velo}^{cam} \in \mathcal{R}^{1 \times 3}$ is the translation vector: velodyne \rightarrow camera

Values for both these matrices are provided with the KITTI database. Using the transformation matrix velodyne coordinates are calibrated with the reference camera. In this dataset, cameras are used to capture images in forward direction only. So, LIDAR data points with positive x values are projected only to the RGB color image. The fusion of the data from these two sensors (velodyne and camera) gives us a more detail model of the vehicle environment that contains both the coordinates and color information of the points. Figure 2-8 shows the result of fusing the point cloud shown in Figure 2-1 with the RGB image of the same scenario shown in Figure 2-2.

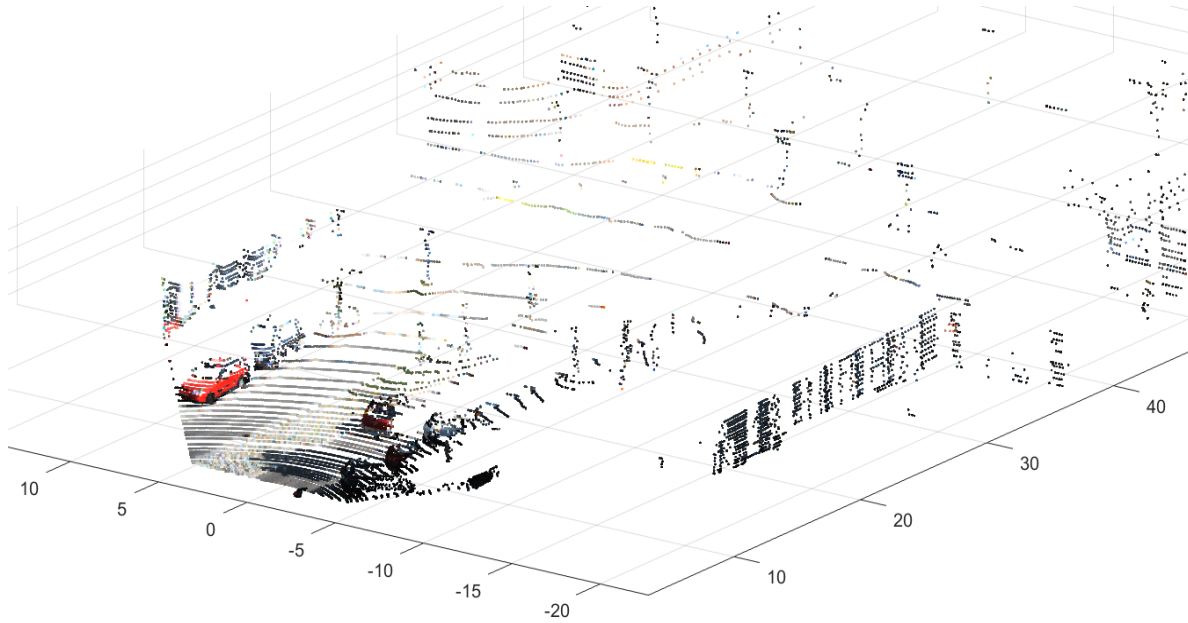


Figure 2-8: 6D model of the scenario shown in Figure 2-1

2.4 Irregularity Detection

Irregularities on road can be defined as some actions of the moving objects that can lead to a collision in near future. Irregular movements can take different forms in varying vehicle environments. On highways, sudden lane change of vehicles and abrupt press of brakes are most common irregularities that may result in serious collisions. In rural roads, wild animals such as deer crossing the roads often causes accidents especially in low visibility situations. In urban environments, irregular events include jaywalking by pedestrians, road sharing by cyclists, careless drivers getting out of parking areas, reckless turning at the stop sign and so on. How to tell those irregularities from regular moving patterns is a fundamental challenge in autonomous vehicles.

In this work, we mainly focus on the irregularities on busy urban roads caused by the pedestrians, cyclists (also motorcyclists), and other vehicles. Figure 2-7 shows an example of irregular movement

of a cyclist (highlighted by red color) sharing the road with respect to the host vehicle (HV). A key observation behind the detection of irregular events from LIDAR data is to predict those moving objects that could result in a potential collision with the HV in the near future [58]. Providing an early warning of irregularity is the objective here; and we propose to tackle this problem in two steps: detect and tracking moving objects using nearest-neighbor search, and irregularity detection from moving objects using linear regression.

2.4.1 Tracking Moving Objects

As irregularities are mostly caused by the moving objects (vehicles, pedestrians etc.), detecting irregularities largely depends on the successful detection and tracking of the moving objects. A robust EM-ICP based moving object detection technique has been described in the previous chapter. In the last stage of this technique, DBSCAN has been used to pack the moving points into some clusters to locate the moving objects. Centroids of these clusters will be used to track those objects in the later frames.

To track the trajectories of the moving objects through multiple frames, Euclidean distance based nearest neighbor search technique has been exploited on the centroids of the moving clusters detected using DBSCAN [57]. Two centroids from adjacent frames are considered as the centroid of the same object if the distance between these two points lies between two empirically set thresholds d_h and d_l . If the calculated distance is larger than d_h , two clusters are not the same object. Again, if the distance is less than d_l , there is no actual movement happened for this object. Only if the distance is between these two thresholds, two clusters are recorded as the same object and a successful tracking operation is declared. This operation is performed for the next couple of frames and the changes in the positions of the moving clusters are used to derive the motion trajectories of

the moving objects. In Figure 2-7, moving cyclist (shown in red color) has been detected and tracked through multiple frames. Four red colored clusters represent the positions of the cyclist in four consecutive frames. Figure 2-9 shows the tracking results of multiple moving objects in the same scenario. Here, four moving objects (different colors and labels) have been detected and tracked. Motion trajectories can be derived from the detected moving clusters and irregularities can also be predicted using these trajectories.

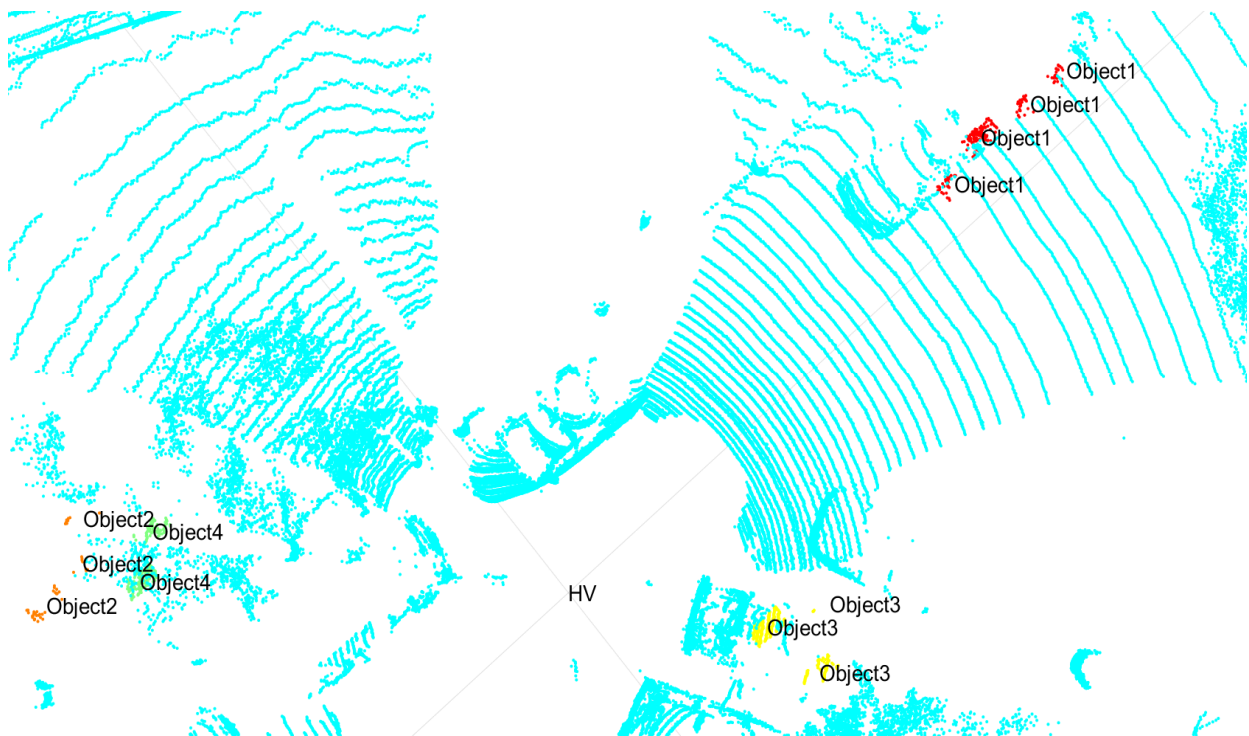


Figure 2-9: Tracking multiple moving objects from LIDAR data

2.4.2 Predict Irregularities using Linear Regression

In a previous work [58], a regression based approach has been proposed to understand the motion trajectories of walking pedestrians. In this experiment, we have used a similar technique to predict

the trajectories of not only pedestrians but also any detected moving objects. Linear regression has been exploited over the centroids of the moving clusters to derive the probable trajectories. After the processing of each new frame, the positions of the centroids of the moving objects are updated and their moving trajectories are recalculated. A critical region A_{cri} is set around the host vehicle based on the width and length of the vehicle. The length and width of this critical area are denoted as L_{cri} and W_{cri} and are evaluated as,

$$\begin{aligned} L_{cri} &= 2L_v + \eta d_f \\ W_{cri} &= 2W_v \end{aligned} \dots \dots \dots (2-3)$$

where, L_v and W_v are the length and width of the host vehicle (HV) respectively. d_f is the distance covered by the HV during the interval of two consecutive frames and η is a tunable constant. Note that the position of the LIDAR sensor is in the center of this critical region.

As the HV is also moving, this critical region will change its position with times. If the motion trajectory of any moving object touches or passes through the critical region at any point, then there exists a possibility of collision if any of the HV and moving object does not change its direction. There is also a region of interest (ROI) based on which, the movements of the detected moving objects are monitored. Objects from outside of the ROI, are not considered as a threat. The moment that an object enters the ROI, it is tracked and the trajectory is derived. The length L_{roi} and width W_{roi} of the ROI are also dependent on the dimensions of the HV. The values of these parameters also vary with the speeds of the vehicles on the roads, e.g. in highways where the speed limits are high, the ROI should be larger than the urban roads with low speed limits. The value of the dimensions of the ROI are calculated using following equations,

$$\begin{aligned} L_{roi} &= 2\mu L_v \\ W_{roi} &= 2\mu W_v \dots \dots \dots (2-4) \end{aligned}$$

Where, μ is a constant that can be varied based on the characteristics of the roads (e.g., varying speed limit and whether there is the presence of traffic lights).

In short, the proposed strategy of irregularity detection can be summarized as follows. When an object enters the ROI, its motion trajectory is derived using linear regression. Then some possible future positions of this object are predicted based on this trajectory. Meanwhile, the critical region around the HV is also tracked and monitored (assuming a constant velocity). If their predicted trajectories intersect with each other, then a possible collision between them could occur in future and the movement of the object is labeled as irregular. Otherwise the event is treated as regular and it is safe for the HV to follow the original motion trajectory. The result of this proposed method has been shown in Figure 2-10, where irregular movement of a cyclist is detected. Positions of the cyclist in different frames are shown using red color and the straight line through the red color clusters is the trajectory of its motion. Blue dot is the projected location of the cyclist at the time of collision if it continues to follow the current trajectory. Rectangular box with broken lines is the projection of the critical region of HV at the collision time, and solid rectangle is the current location of HV. The word ‘Critical’ is shown in the current location of the cyclist which indicates the necessity of changing the current trajectory of the vehicle.

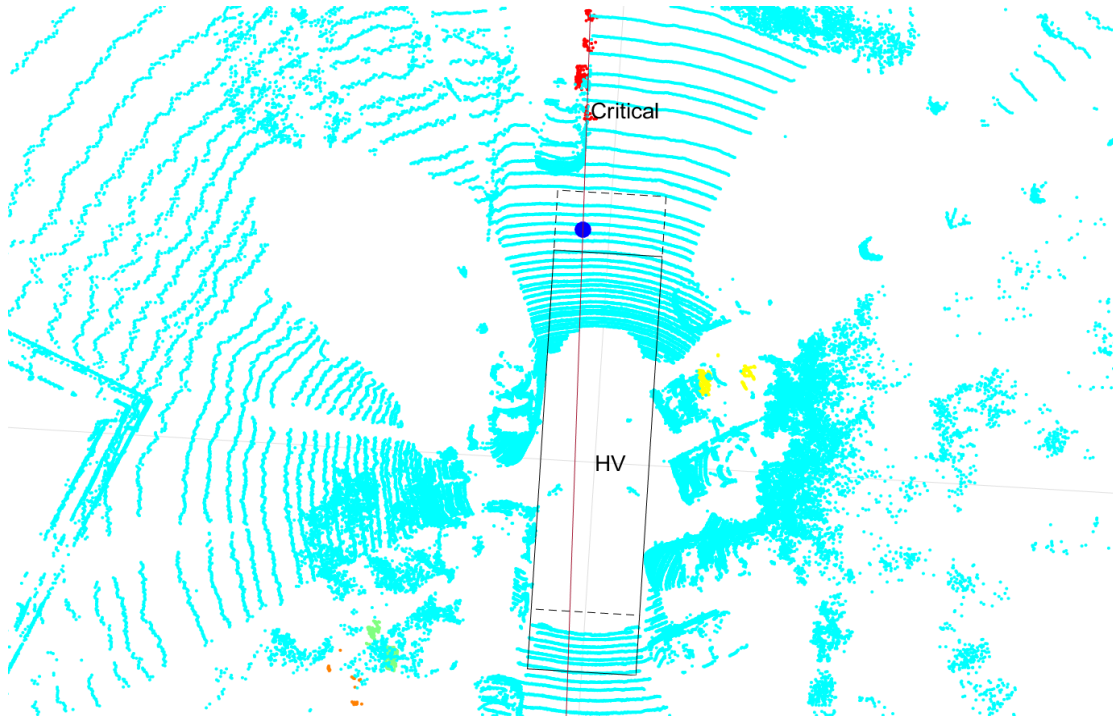


Figure 2-10: Predicted irregular movement of a cyclist (red color object)

2.5 Vehicle Environment Understanding

In the previous sections, we have discussed about vehicle environment reconstruction based on fusion of 3D point cloud data and RGB color image (which gives a six-dimensional representation of the environment). An approach to detect irregularities from extracted moving objects has also been addressed. In this section, we will focus on the understanding of different objects in the reconstructed environment. As, vehicles and pedestrians are the most important components on the roads, prime objective of this section is to present approaches to detect pedestrians and vehicles. For both cases, histogram of oriented gradient (HOG) [33] based object detection algorithms have been proposed. Therefore, this section is organized in two different parts – first one describes HOG based pedestrian detection technique and the other part describes a Position and Intensity – included HOG (PIHOG or π HOG) based approach for vehicle detection from the reconstructed environment.

2.5.1 Pedestrian Detection from Point Cloud

For detecting the pedestrians, only 3D point cloud has been used instead of 6D XYZ+RGB map. Because of the nature of human body shape, color information of a pedestrian in point cloud often get corrupted by the background color, which makes the pedestrian difficult to distinguish. The proposed method has been conducted in two steps: First, DBSCAN is applied to cluster the points into some groups to locate all the different objects and distance map (d-map) is generated for each object. Second, HOG features are extracted from the d-maps and a supervised support vector machine (SVM) [59] is used to recognize the pedestrians.

2.5.1.1 Clustering and D-map Generation

In a typical point cloud, XYZ coordinates of each point are given. But, the information of actual sources of the points (i.e. which point is part of which object) is absent. To achieve this information, density based clustering DBSCAN [57] is exploited. As the points from same object are in very close distances from each other, DBSCAN packs these points in the same cluster, i.e. each resulting cluster represents a different object. Clusters with very few points ($n \leq 5$) are considered as outliers. To improve the clustering result, points from the ground level (i.e. road surface) are removed before applying DBSCAN based on the Z-values of the points (shown in 2.2.2.1). Figure 2-11(b) shows the result of DBSCAN clustering applied on the point cloud scenario shown in (a). Different clusters are shown using different colors and numbers. These clusters represent different objects detected in this scenario.

Next step is to take each cluster separately and generate 2D distance-map (d-map) for each cluster. D-map consists of the Euclidean distances ($d = \sqrt{x^2 + y^2 + z^2}$) of each point from the center (0,0,0) of the point cloud coordinate. As the point cloud coordinate is measured based on the location of the velodyne sensor, d-map basically gives the distances of the points from the sensor. If the projection of a point P(x,y,z) onto the two dimensional d-map is D(r, c), then the location of D is measured using following equations,

$$\begin{aligned} r &= I_r + (C_z - z) \times N_{ppm} \\ c &= I_c + (C_y - y) \times N_{ppm} \end{aligned} \dots \dots \dots (2-5)$$

Where, (C_x, C_y, C_z) is the centroid of the cluster, (I_r, I_c) is the location of center pixel in the d-map. N_{ppm} is the number pixels per meter, i.e. the number of pixels used in d-map to cover one meter in point cloud (in this experiment, $N_{ppm} = 32$). Figure 2-12 shows some sample d-map representations of both human and non-human clusters. These generated d-maps are then used to extract HOG features.

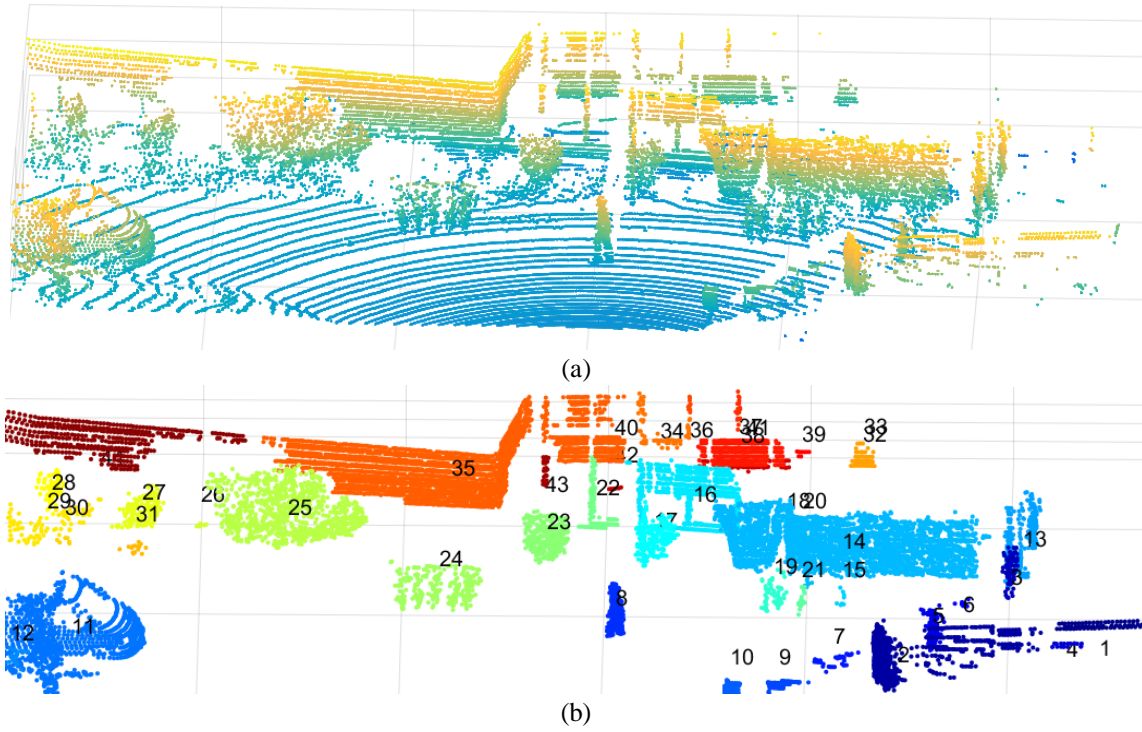


Figure 2-11: Clustering result- (a) original point cloud, (b) clusters are shown in different colors and numbers

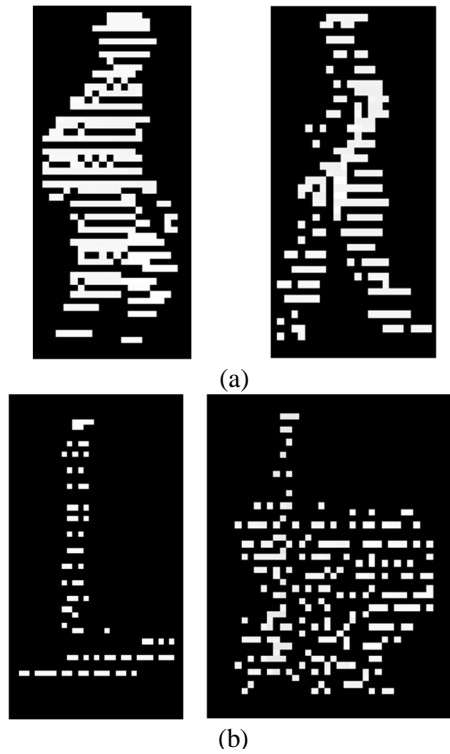


Figure 2-12: 2D d-maps of (a) pedestrian, and (b) non-human clusters

2.5.1.2 HOG-SVM Based Pedestrian Detection

The Histogram of Oriented Gradient (HOG) based human detector is one of the most popular and successful “human detectors”. Dalal and Triggs introduced it in the article [33]. HOG uses “global” feature to describe a human rather than a collection of “local” features, i.e. the image of an entire human can be represented by a single feature vector, as opposed to many feature vectors representing different parts of the human body. For a typical image, the procedure of the HOG feature extraction starts with a sliding detection window that is moved around the image. But, in this experiment, we have applied HOG to the d-map image of each cluster separately. Therefore, all the d-map images are reshaped to be of same dimensions, but no sliding window is used. For each image, a HOG descriptor is computed. This descriptor is then tested using a supervised classifier which classifies it as either “human” (i.e. pedestrian) or “non-human”.

In this proposed method, the common resolution of all d-map images is 32x64 (32 pixels wide, 64 pixels high). First, the gradient vector $[g_h \ g_v]^T$ is computed at each pixel (both magnitude and angle) of this image, where, g_h and g_v are the gradient of intensities in horizontal and vertical directions respectively. This 32x64 image is then divided into some 8x8 blocks with 50% overlapping. Further, each block is divided in to four 4x4 cells. Then, the gradient vectors in each cell were put in a 9-bin (0~180 degrees) histogram. L2 normalization method is used for normalizing the histogram to make it invariant to the illumination change. The feature vector size can easily be calculated from the above numbers. The 32x64 image is divided into 7 blocks horizontally and 15 blocks vertically, for a total of 105 blocks. Each block contains 4 cells with a 9-bin histogram for each cell, for a total of 36 values per block. This brings the final feature vector size to 7 blocks horizontally x 15 blocks vertically x 4 cells per block x 9-bins per histogram = 3,780.

HOG feature vectors are extracted from the manually labelled positive (human) and negative (non-human) d-map images. These feature vectors are then used to train a support vector machine (SVM) [59] classifier with linear kernel. Support Vector Machine (SVM) is a method used for the classification of both linear and nonlinear data [60]. A nonlinear mapping is used to transform the original training data into a higher dimension, and within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a “decision boundary” separating the data points of one class from another). If an appropriate nonlinear mapping (to a sufficiently high dimension) can be produced, data from two classes can always be separated by a hyperplane. Support vectors (“essential” training data points) are used to find this hyperplane. SVM can be of linear and nonlinear kernels. In our experiment, linear SVM has been used to classify two classes (human and non-human) due to its low complexity. This trained SVM is later used to detect pedestrians in test point cloud scenario.

So, when a point cloud frame is tested, first its ground level points are removed, and all the objects are separated using DBSCAN clustering. Then, d-map image is generated for each cluster based on the Euclidean distances of the points from the sensor location. HOG features are extracted from these two-dimensional images and a trained SVM is used to classify the images (to check whether the corresponding cluster is pedestrian or not). Figure 2-13 shows the result of applying HOG-SVM based pedestrian detection technique to the point cloud shown in Figure 2-11.

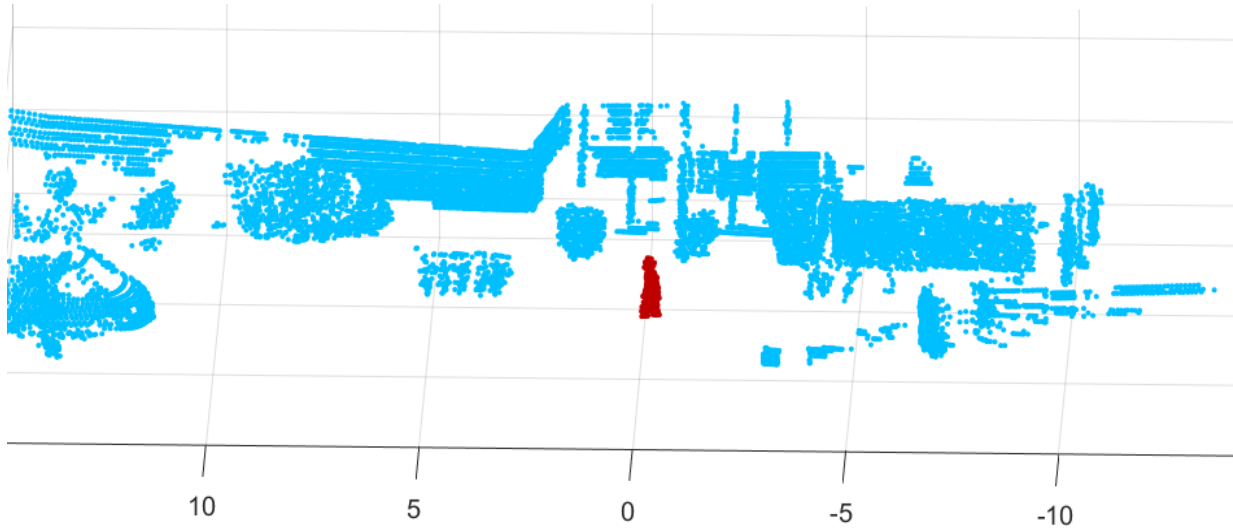


Figure 2-13: Detected pedestrian (red colored) in the scenario shown in Figure 2-11

2.5.2 Vehicle Detection using PIHOG

Similar to pedestrian detection, HOG based technique can also be applied to detect vehicles. But, in case of vehicle we have used both point cloud and color information (XYZ + RGB), i.e. the fused 6-dimensional representation described in section 2.3. As, the appearance of a vehicle is more stable than the pedestrian in point cloud, detection of vehicle using color information along with XYZ data is not prone to the interference of the background. Again, because of the large and flat shape of the vehicle, some drawbacks of the regular HOG based techniques need to be resolved before applying. Two drawbacks are mainly taken under consideration [47]:

- 1) HOG is basically a histogram; the locations of the gradients are not stored. As a result, two completely different cells may have the same histogram. This problem has been explained in Figure 2-14. Here, two cells have 16 pixels, and the gradient vector of each pixel has been illustrated using the red arrow. For simplicity, all gradients are assumed to have same

intensities. Although, the two cells have different gradients, they have the same histogram of oriented gradients because both cells have the same number of 45° and 135° components.

- 2) Typical HOG features contain only edge information of an image; intensity information is not used in HOG. Only edge based features are not sufficient in case of vehicle detection.

From the above two points it is clear that, there remains opportunity to further improve the performance of vehicle detection by using the image intensity and the location of the gradients. Based on these two ideas, Position and Intensity – included HOG (PIHOG or π HOG) has been introduced in [47] to detect vehicle in RGB images. In this experiment, PIHOG has been exploited on the 6D representation (XYZ+RGB) of the vehicle environment.

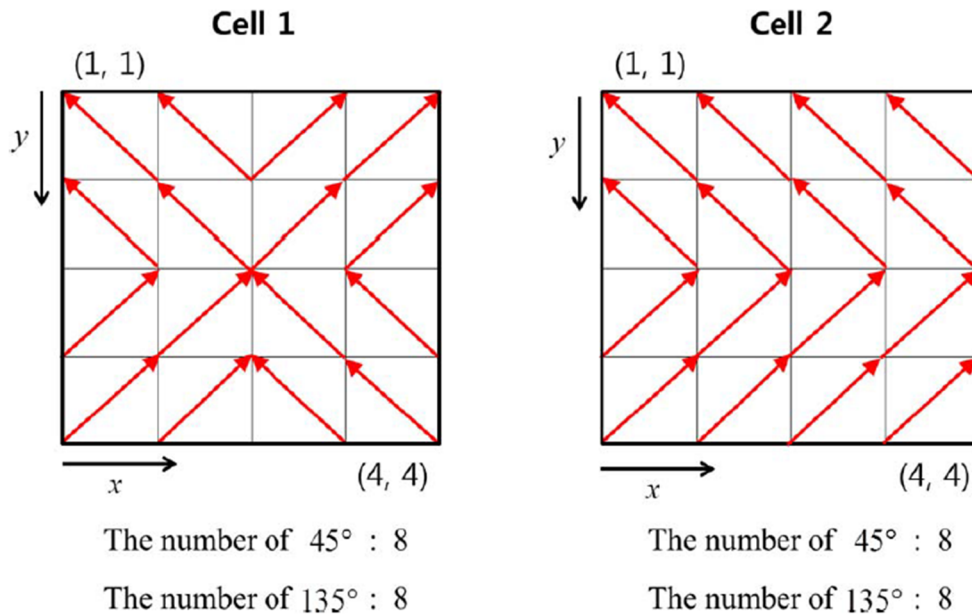


Figure 2-14: Two different cells with same histogram of gradients [47]

Before extracting PIHOG features, the 6D map needs to be transposed into a 2D image. In this experiment, front view of the map has been used. The basic idea is similar to D-map generation described in 2.5.1.1, except in this case RGB color information of the points is used to generate the

2D image instead of distance. Also, no clustering algorithm has been used to separate the objects. Figure 2-15 illustrates the 2D front view of the vehicle environment shown in Figure 2-8. The goal is to develop a technique that will be able to detect the vehicles (can clearly be observed in the figure) using PIHOG-SVM combination. The main advantage of using this fused data instead of RGB image only is to solve the scaling problem. In regular RGB image, all the objects face this scaling effect, i.e. nearer object appears in larger shape, and the appearance of an object gets smaller with the distance. But, as the 6D map gets the height information from the z-values of the point cloud, objects from different distances appears to have same heights in the 2D transposed image.



Figure 2-15: 2D front view of the 6D map shown in Figure 2-8

2.5.2.1 PIHOG (or π HOG)

Position and Intensity – included HOG (PIHOG, also written as π HOG) was introduced by Kim et al. [47]. To solve the two drawbacks of HOG mentioned in the previous section, PIHOG feature has been realized. Beside the regular HOG features it includes the information of the gradient locations and pixel intensities. These two extra features have been described in the following subsections.

2.5.2.1.1 Position Features

As illustrated in Figure 2-14, two cells with different gradient vectors can have same histogram (in other words, same HOG feature). To make the features more distinguishable, the positions of the oriented gradients are taken into consideration as new features. The procedure of extracting this feature has been described in this section. Let $\theta(x, y, c)$ be the orientation of the gradient at (x, y) position of the c cell ($0 \leq \theta \leq 2\pi$). The orientations of the gradients are quantized in T bins and the expression of the orientation bin $B(x, y, c)$ is given by,

$$B(x, y, c) = \left\lceil \frac{T\theta(x,y,c)}{2\pi} \right\rceil, \quad 0 \leq \theta \leq 2\pi, \quad B(x, y, c) \in \{1, \dots, T\} \dots \dots \dots (2-6)$$

Then the means of x and y positions of the d -th bin has been calculated using the following expressions,

$$M_{x,d}^c = \frac{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} x \Pi[B(x,y,c)=d]}{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} \Pi[B(x,y,c)=d]}, \text{ and } M_{y,d}^c = \frac{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} y \Pi[B(x,y,c)=d]}{\sum_{x=1}^{c_x} \sum_{y=1}^{c_y} \Pi[B(x,y,c)=d]} \dots \dots \dots (2-7)$$

Where, c_x and c_y are the width and height of the cell respectively, $\Pi(\cdot)$ is an indicator function that returns '1' if the argument is true, otherwise, returns '0'. After calculating the means for all the orientation bins in c -th cell, we get the new features $P_c = [M_x^c, M_y^c]$ (where, $M_x^c = [M_{x,1}^c, \dots, M_{x,T}^c]$ and $M_y^c = [M_{y,1}^c, \dots, M_{y,T}^c]$) that deals with the computation of the position part.

2.5.2.1.2 Intensity Features

In this section, intensity invariant region (IIR) based features are proposed. All the positive vehicle images are used to extract the IIR. Let $V = \{s_1, s_2, \dots, s_{N_v}\}$ be the set of all positive vehicle images and $s_i = [s_{i1}, s_{i2}, \dots, s_{iN}]^T$ be the i -th positive vehicle image; where, s_{ij} is the intensity of the j -th pixel in the i -th vehicle image. N is the total number pixels in each image, and N_v is the size of the

set of positive images. In this experiment, all the sample images (both positive and negative) are resized in to 48x48 images ($N = 2304$). To determine the IIR across the positive vehicle images, the mean and standard deviation of the images are calculated using the following equations,

$$M = \frac{1}{N_v} \sum_{i=1}^{N_v} s_i = [m_1, m_2, \dots, m_N]^T \dots \dots \dots (2-8)$$

$$\sigma = \sqrt{\frac{1}{N_v} \sum_{i=1}^{N_v} (s_i - M) \circ (s_i - M)} = [\sigma_1, \sigma_2, \dots, \sigma_N]^T \dots \dots \dots (2-9)$$

where \circ denotes component-wise multiplication. The main idea of IIR is that, the regions with low standard deviations correspond to the regions where all the positive vehicle images have similar intensity values. Thus, intensity values in these regions are considered to be unique for the vehicles regardless of the colors of the vehicle. These regions are called IIR. For the purpose of feature extraction, values of the standard deviation image are divided into K number of intervals. If the range of the values of k -th mask U_k is given by $[\varepsilon_k, \varepsilon_{k+1}]$, then the mask can be described by

$$U_k = \{j | \varepsilon_k \leq \sigma_j \leq \varepsilon_{k+1}, j = 1, 2, \dots, N\} \dots \dots \dots (2-10)$$

Figure 2-16 shows the mean image (a), and the standard deviation image (b) of the positive vehicle images. In our experiment, we have used 20 intervals (i.e. $K = 20$). For feature matrix generation, only first five masks ($n = 5$) have been considered, as low standard deviation values indicate the region of similar intensities for all the vehicles. Now, for a test image s , standard normal deviate image is computed as,

$$z = \left(\frac{1}{\sigma}\right) \circ (s - M) \dots \dots \dots (2-11)$$

Applying the first n IIR masks ($U_1, U_2, \dots, U_n; n = 5$) on z , the features correspond to the IIR masks can be computed by,

$$h_k = \frac{1}{|U_k|} \sum_{j \in U_k} z_j \dots \dots \dots (2-12)$$

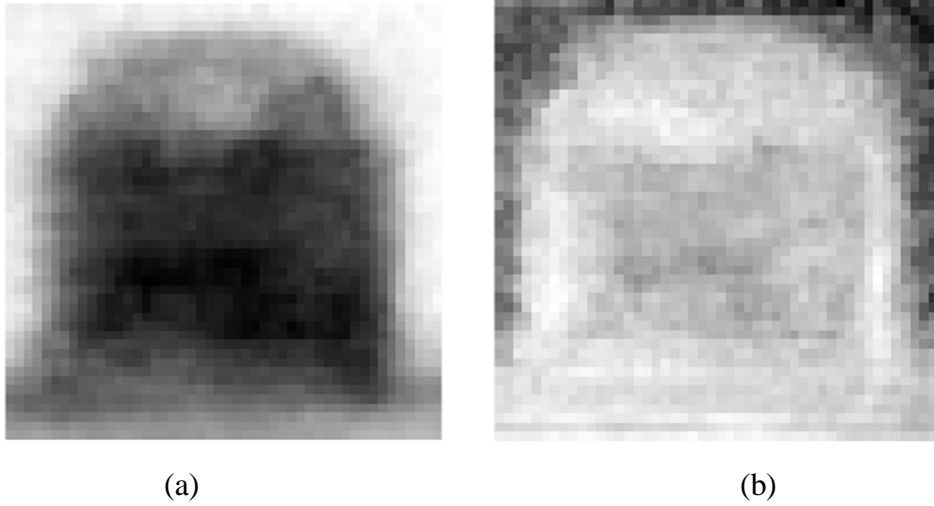


Figure 2-16: (a) Mean, and (b) standard deviation all positive vehicle images

2.5.2.1.3 Formation of PIHOG Features

The procedure of PIHOG starts like the conventional HOG. A sliding window (40x40 for this experiment) is used to take the image segments and the segment is reshaped to size 48x48. Each segment is decomposed into C cells ($C = 4$), and HOG features are extracted from each cell. The gradients in a cell are assigned to T bins ($T = 9$), i.e. dimension of extracted HOG features from each window is CT . The position part of PIHOG P_c (for a cell) is a $2T$ dimensional vector (T values in each of x and y direction). Hence, for a window position part of PIHOG has $2CT$ dimensions. The intensity part of PIHOG in a window consists of n dimensional IIR feature vector h_k ($k = 1, 2, \dots, n$). After combining all the features, final feature matrix of PIHOG for a window is defined as,

$$F_{\pi HOG} = [F_{HOG}, P_1, \dots, P_C, h_1, \dots, h_n] \dots \dots \dots (2-13)$$

where, F_{HOG} is the conventional HOG feature. The total length of the feature matrix is $3CT+n$. PIHOG features are extracted from both positive (vehicle) and negative (background, i.e. road, trees, buildings etc.) samples. Like pedestrian detection, a linear support vector machine (SVM) has been

trained using the PIHOG features and later used to detect vehicle from test images. The result of PIHOG-SVM based vehicle detection technique (applied to the image in Figure 2-15) has been shown in Figure 2-17.

Like all other sensors, both LIDAR and camera have distance limitations. An autonomous vehicle can reconstruct the map of neighboring environment up to a certain distance using these local sensors. To extend the map, dedicated short-range communications (DSRC) based vehicle-to-vehicle (V2V) communication can be exploited to share the locally generated among the neighboring vehicles.



Figure 2-17: Detected vehicles (in red boxes) using PIHOG-SVM

Chapter 3: Power Control in V2V

3.1 V2V Communications System

Vehicle-to-vehicle (V2V) communications system is a technology that forms a wireless ad hoc network on the roads to allow autonomous vehicles to communicate with each other. Such networks are also known as vehicular ad hoc networks (VANETs). Via V2V, Vehicles can communicate each other with information of neighboring environment, emergency warning, intersection collision warning, cooperative cruise control etc. In this experiment, DSRC based V2V scheme has been used for the transmission of locally sensed (RADAR-sensed) neighboring environment information and thus generating a more complete and extended map of the neighborhood.

3.1.1 Dedicated Short Range Communication (DSRC)

Dedicated Short-Range Communications (DSRC) provides a means for low latency communication link with 75 MHz of spectrum at 5.9 GHz band allocated by Federal Communications Commission (FCC). In this experiment, the main goal of using the vehicle-to-vehicle (V2V) communication is to ensure safe, interoperable connectivity so that all the vehicles can achieve the knowledge of the environment beyond the range of the local sensors. To facilitate this, V2V communications require a safe and secure wireless interface that can communicate within a short delay and perform well even in harsh environments. DSRC is specifically designed to facilitate all these requirements. Although DSRC is like Wi-Fi technology, it is preferred over Wi-Fi because the number of devices that use the unlicensed Wi-Fi channels would make significant interferences that could hamper the safety concerns of vehicular applications. Some significant features of DSRC are,

- DSRC allocates a specified spectrum dedicated for vehicular safety applications to enable secure communication.
- It allows vehicle to establish communication in a very short time.
- DSRC is designed for high speed communication (up to 27 Mbps).
- It employs a robust OFDM modulation scheme to cope with severe environments.
- DSRC provides interoperability between different automotive manufacturers using standardized protocols at different layer

We have used DSRC based V2V communications to facilitate an interoperable connectivity among the neighboring vehicles. The performance of this communication scheme depends on the transmitted message size, transmission rate, and transmission power. Article [27] proposed an adaptive message length and content control algorithm for cooperative vehicle system. In our experiment, we have studied a power control technique coupled with this content control algorithm.

3.2 System Model and Communication Architecture

Figure 3-1 shows a general model of a cooperative automated vehicle system. In this system, each vehicle contains a set of local sensors such as RADAR, LIDAR, cameras etc. to collect information of the neighborhood and create a real-time three-dimensional map of the surroundings of that vehicle up to the sensing distance of the equipped local sensors. In advanced vehicular system, safety applications such as automated crash-avoidance system, pre-collision alert, lane-keep assist, or adaptive cruise control system can utilize this map information gathered by local sensors, as these kind of safety applications depend only on the information regarding the immediate neighborhood. But some other higher-level applications such as long-term trajectory planning, and automated driving applications require a larger map than the locally sensed map. For example, a locally sensed

map can give information of a neighboring car, but cannot predict the information regarding the next intersection that the vehicle will pass within next few seconds.

Exchange of the locally sensed map information among the vehicles can be the solution to the problem to extend the map. Each vehicle contains its own local map generated using locally sensed information. If one of the vehicles broadcasts its own local map, it would be useful to other vehicles to include the new objects (unknown to their local sensors) to their own maps. A complete map of the surroundings can then be generated by combining the observations through local sensor collections and the information received through communication channel from other vehicles. The information extracted from this updated map will then be broadcast to the neighboring vehicles within the communication range. In this method, the coverage map can be expanded so that the automated driving applications can get more accurate information. Again, based on the vehicle density, the size of the map information to be exchanged among the vehicles can be very large and it can easily lead to the congestion in the network of DSRC transceivers. So, a scalable system is required which can handle the congestion without affecting the accuracy.

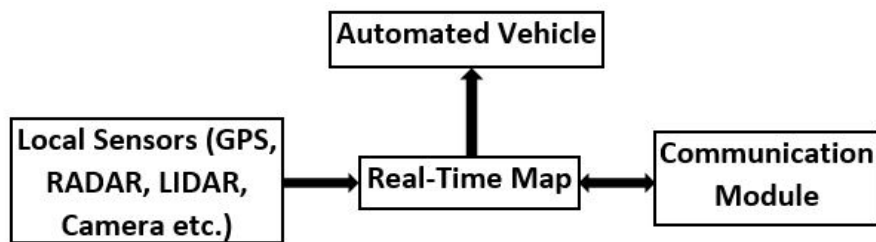


Figure 3-1: Cooperative automated vehicle system model

To realize the inter-vehicle communication scalability problem, we have considered the simulation environment studied in [22]. A 4-km long highway shown in Figure 3-2 is considered to depict the

vehicle movement. It contains three lanes in both directions and a winding section in the middle in which the radius of the curvature is 40 meters. The average speeds of the cars within the three lanes in each direction are 17 m/s, 18 m/s, and 19 m/s respectively from the rightmost to the leftmost lane. Mobility model is generated using traffic simulator SUMO with the modifications described in [22]. For propagation loss model, three-log-distance propagation path loss model [61] and Nakagami fading model are used. Payload size is 84 bytes, i.e. to include the information of an object in the map, 84 bytes are required.



Figure 3-2: Road topology used in the simulation [22]

In cooperative vehicle system, communication among the vehicles play the majority roles to extend the map information, rather fully rely on the local sensing modalities. In this work, as communication architecture, the method proposed in [27] is utilized. This method proposed to enhance the map created using local sensors by collecting information from other vehicles through communication channel. This updated map is also disseminated to other vehicles. This model follows the operation model for cooperative vehicle-safety system described in [23]. Figure 3-3 shows the functional diagram of the communication architecture in this work.

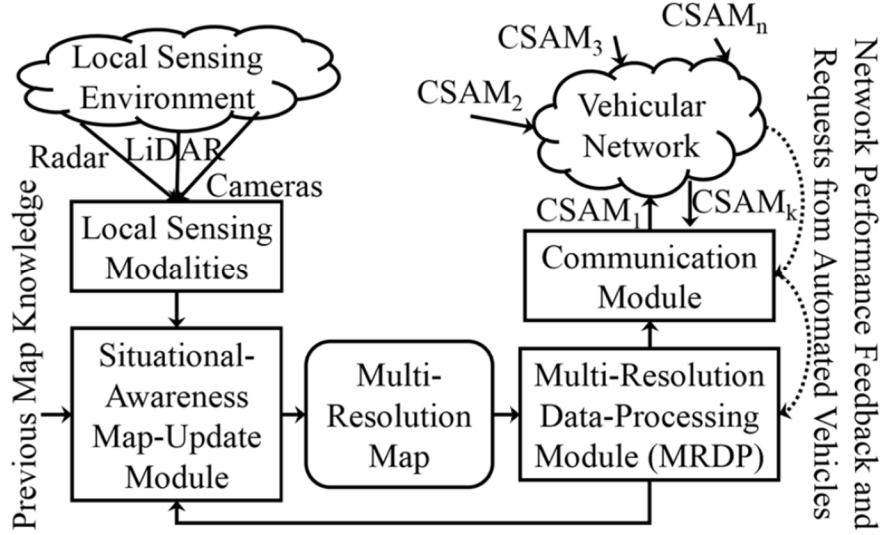


Figure 3-3: Scalable communication architecture proposed in [27]

3.3 Message Length and Content Control Scheme

For adaptive content control, a distance-based content (DBC) control scheme proposed in [27] is used, where a general communication framework was proposed through which vehicles exchange information from multi-resolution maps generated using local sensory data (i.e. camera and LIDAR). In this approach, a descent-based message-size control algorithm [62] has been exploited that updates the optimal message size at iteration $t + 1$ using:

$$L_{OPT}^{t+1} = L_{OPT}^t + \eta \times (CBR^* - CBR^t) \dots \dots \dots (3-1)$$

where L_{OPT}^t represents the message size at iteration t , CBR^t is the resulting observed channel busy ratio, and CBR^* is the desired channel busy value for maximum throughput. It is shown in [21] that throughput is maximized when CBR is at 0.68, so in this case $CBR^* = 0.68$. Here, η is a tunable gain that can be adjusted to converge the algorithm. The optimal message is bounded between two limits L_{max} and L_{min} . The maximum message size should not be higher than L_{max} , even if CBR is low

enough to support higher message size. L_{min} is the minimum message size required to represent an arbitrary object in the map, i.e. to transmit a message, a vehicle should have the information about at least one object in the map.

For the selection of message content, a distance r_0 is set as a must include distance, i.e. DBC includes objects with less distance to transmitter than r_0 with probability 1. Beyond this distance, for an object located at distance r , the probability of inclusion is calculated by,

$$\Pr(ObjectInclusion) = \begin{cases} 1 & \text{if } r \leq r_0 \\ \lambda e^{-\lambda r} & \text{if } r > r_0 \end{cases} \dots \dots \dots (3-2)$$

where, $\lambda = \frac{1}{r_0} \ln \frac{1}{|1-r_0|}$ and $0 \leq r_0 \leq 1$.

The packets formed by DBC are then transmitted using different transmission powers and position tracking errors (PTE) are measured to analyze the effect of transmission power on map accuracy. For further analysis, channel busy ratio (CBR) and packet error rate (PER) are also observed for those transmission powers. Based on the analysis, a power control scheme is proposed, which operates in combination with the content control algorithm to give an optimum position tracking accuracy.

3.4 Proposed Power Control Scheme

In this work, the effect of the changes in transmission power is observed and a power control algorithm is proposed based on the observation results. The distance dependent content control scheme, discussed in the previous section, is exploited as the congestion control algorithm. In this section, effects of changing transmission power have been studied and the power control scheme has been described.

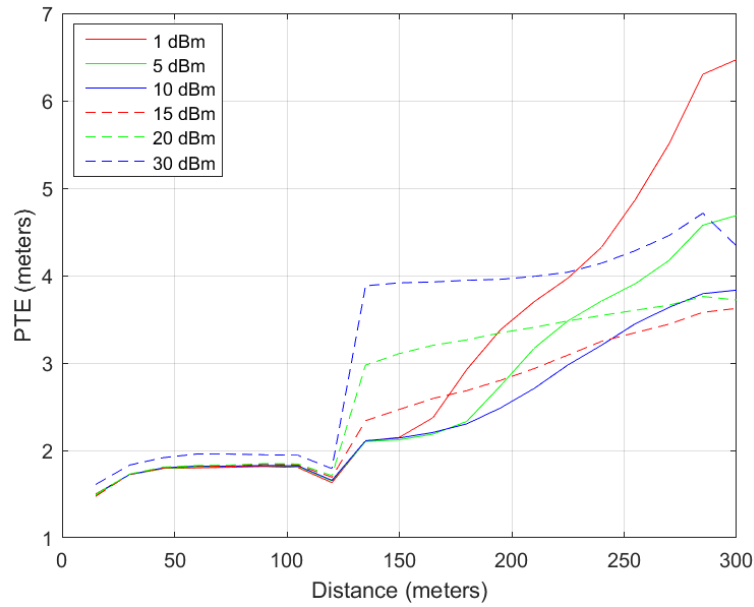
3.4.1 Effect of Transmission Power on Map Accuracy

Map accuracy is evaluated based on the actual position tracking error (PTE), i.e. difference between mapped-distance (using both from local sensors and communicated information) and actual sender-receiver distance provided by the simulator mobility model. Map accuracy is inversely proportional to PTE, i.e. if the PTE can be kept low, more accurate map can be achieved.

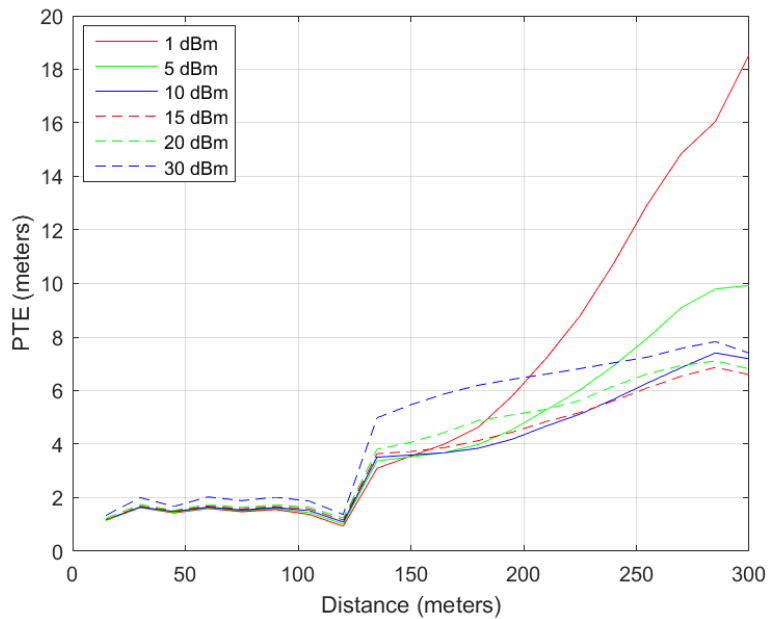
We performed NS-3 simulations for densities of 125 vehicles/Km and 250 vehicles/Km, and calculated 95 percentile PTE for six different transmission powers (1, 5, 10, 15, 20, and 30 dBm). In the simulation, radar-sensed data are used up to 120m distance and beyond this distance, communication data are used to collect map information. Figure 3-4(a) and (b) shows the position tracking error (PTE) plots for 125 vehicles/Km and 250 vehicles/Km respectively using the transmission powers mentioned above. From the figures, we can see that PTE decreases with the increment of power until 10 dBm, and begins to increase again after that. That means, accuracy of the generated map reduces in higher powers.

To understand this result we need to look into the information age (IA) vs distance plots for the same transmission powers and vehicle densities shown in Figure 3-5. It shows that for high powers (20, 30 dBm) the information received by a vehicle is older than for low powers, which results in low accuracy. For a given loss model, packets transmitted with higher transmission power, travel to longer distance, which causes more congestion in the propagation channel, i.e. high channel busy ratio (CBR), as shown in Figure 3-6. To keep the channel congestion under a tolerant level (less than 0.68), the adaptive length control algorithm described in previous section has been used. In case of high congestion (for higher transmission powers), this algorithm reduces the packet size (i.e. message length). Figure 3-7 shows the results of message length adaptation process based on the CBR values for different transmission powers. To reduce the message size, less objects are included

in the transmitted packet (i.e. less information is gathered), and the information is updated less frequently, which is the main reason behind the older information for higher transmission powers.

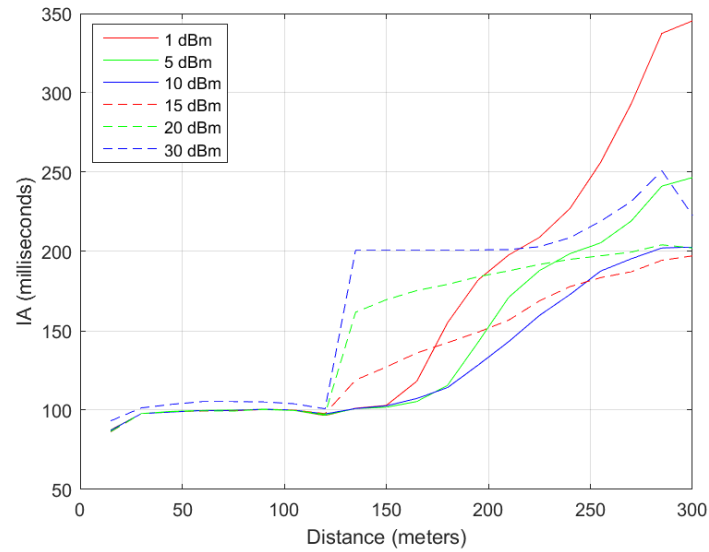


(a)

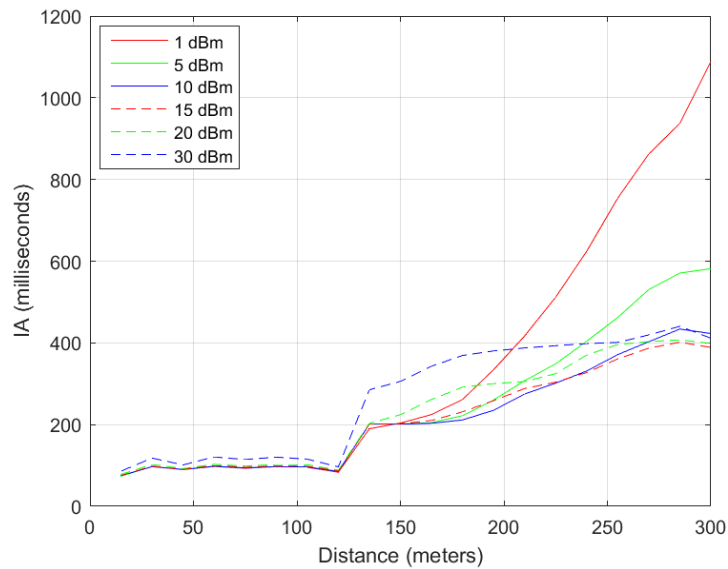


(b)

Figure 3-4: PTE vs distance for (a)125 vehicles/Km, and (b) 250 vehicles/Km



(a)

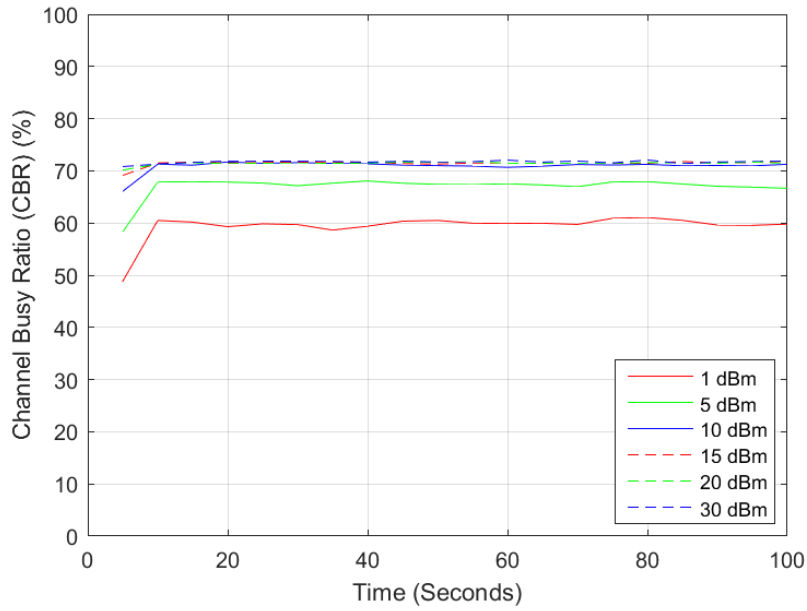


(b)

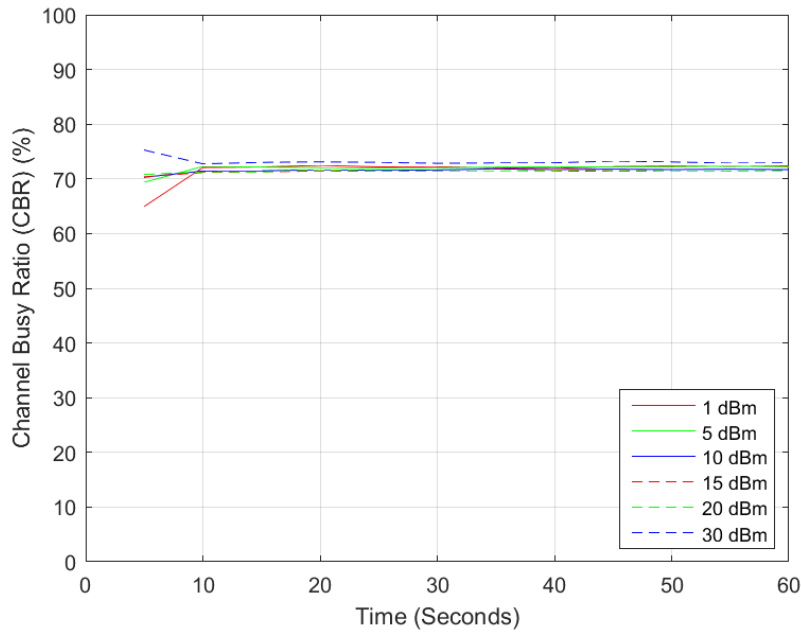
Figure 3-5: Information age vs distance for (a) 125 vehicles/Km, and (b) 250 vehicles/Km

Reduction in packet size also results in low packet error rate (PER). Typically, as more congestion occurs for high transmission power, more packets are expected to be lost, i.e. PER should increase. But, the congestion control algorithm reduces the message length to keep the channel less congested,

which causes the less packet loss. So, unlike PTE, in case of higher transmission powers (e.g. 20, 30 dBm) less PER occurs (Figure 3-8).

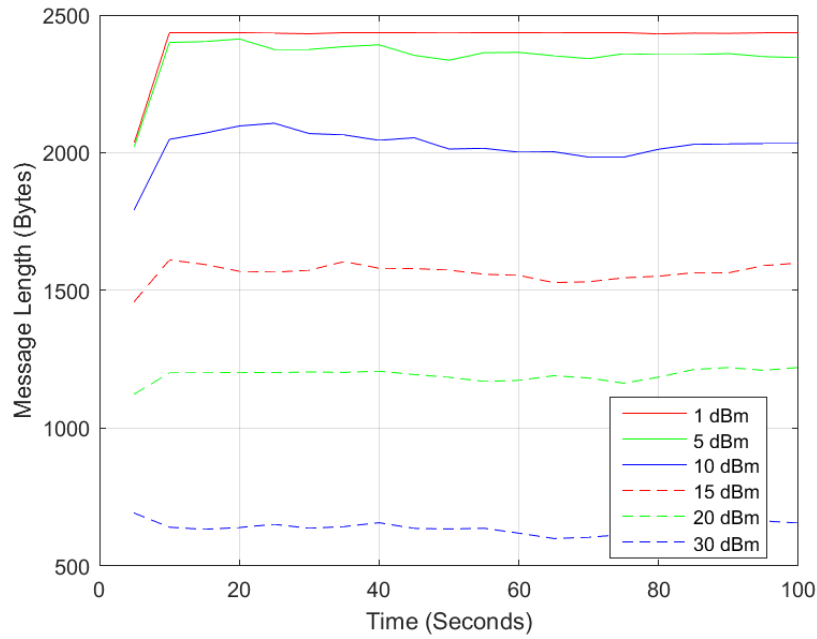


(a)

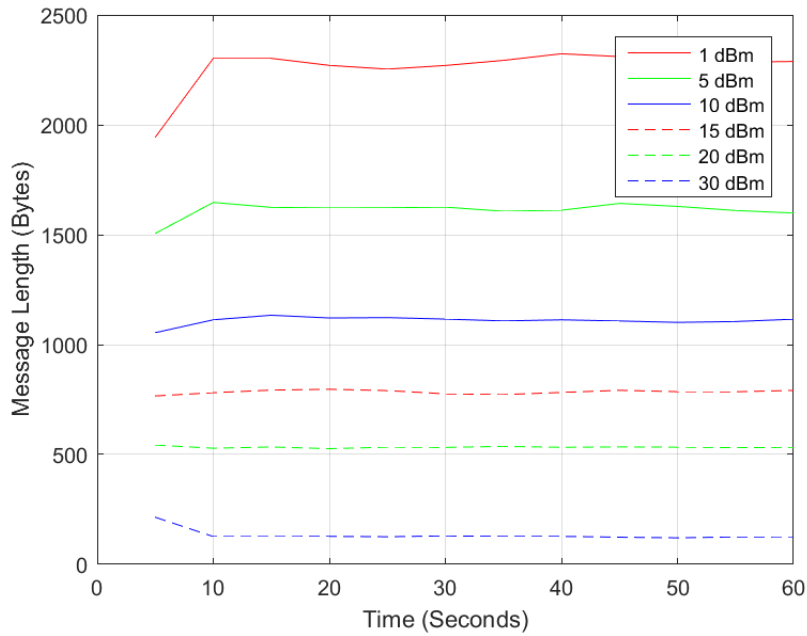


(b)

Figure 3-6: CBR vs time for (a) 125 vehicles/Km, and (b) 250 vehicles/Km

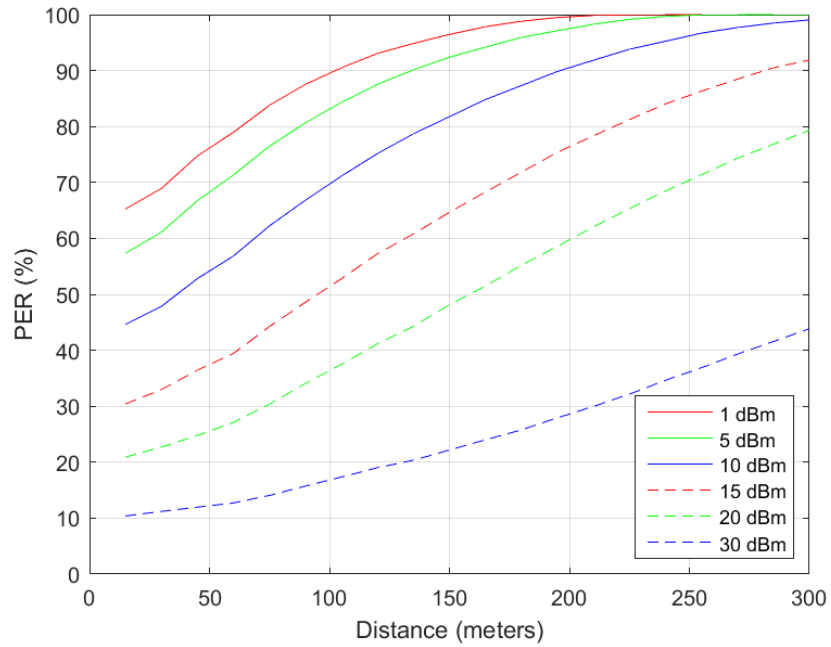


(a)

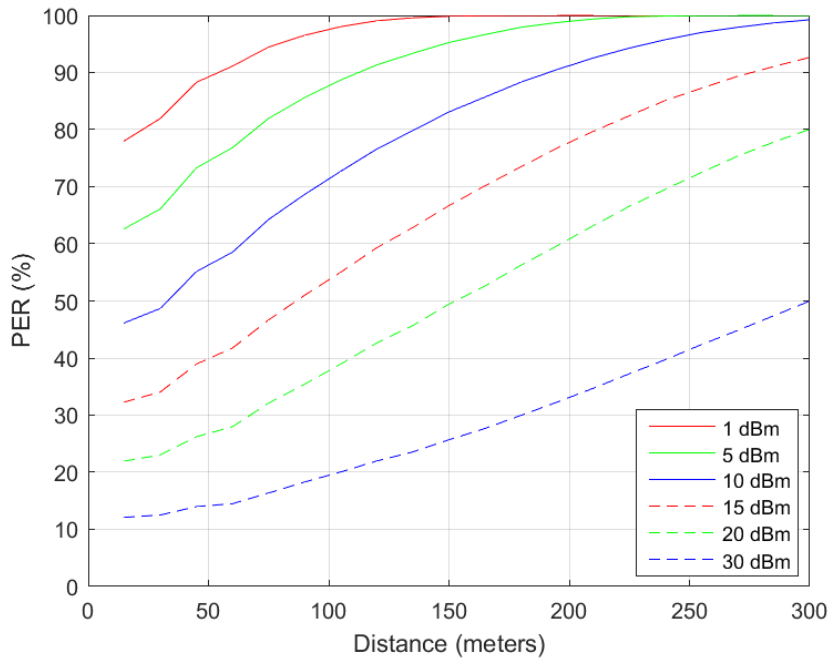


(b)

Figure 3-7: Message length vs time for (a) 125 vehicles/Km, and (b) 250 vehicles/Km



(a)



(b)

Figure 3-8: PER vs distance for (a) 125 vehicles/Km, (b) 250 vehicles/Km

To summarize this discussion, as more packets travel to longer distance if transmitted with high transmission power, PTE increases with the increment of the power. But after a certain level (10

dBm) it begins to increase again because of more channel congestion due to high transmission power. To minimize congestion, message length is reduced that results in less information and hampers the map accuracy. Based on these results we have tried to develop an adaptive power control scheme coupled with adaptive content control algorithm, which can give optimum message length to achieve better map accuracy.

3.4.2 Power Control Strategy

In previous discussion, it is observed that in case of close distance (less than 250m), packets transmitted with 10 dBm transmission power give best result, but beyond this distance 15 dBm gives lower PTE. So, in the proposed power control scheme transmission power is varied between these two power limits ($10\text{dBm} \leq P \leq 15\text{dBm}$). Main goal of this scheme is to keep the message length in between an optimum range ($L_{op,min} \leq L \leq L_{op,max}$) so that the congestion will remain under the tolerant level, and at the same time, enough information can be gathered to generate a more accurate map. The maximum ($L_{op,max}$) and minimum ($L_{op,min}$) values of the optimum range of the message length are found using the following formula:

$$L_{op,min} = \{L_{max} - (d \times \eta_d)\} \times n_B; \quad L_{op,min} \geq L_{min} \dots \dots \dots (3-3)$$

$$L_{op,max} = L_{op,min} + (a \times n_B); \quad L_{op,max} \leq L_{max} \dots \dots \dots (3-4)$$

where, d is the vehicle density, η_d is density factor, n_B is the number of bytes needed to describe an object, and a is a constant to tune the optimum range. NS-3 simulations are conducted for 125 vehicles/Km and 250 vehicles/Km densities, so here d takes the values 125 and 250 respectively. Values used for other parameters are given in Table 3-1. And the transmission power adjustment is done at $t+1$ iteration using:

$$P_{t+1} = \begin{cases} P_t + 1, & \text{if } L > L_{op,max} \\ P_t - 1, & \text{if } L < L_{op,min} \end{cases} \dots \dots \dots (3-5)$$

Table 3-1: Parameters used in simulation to detect optimum

Parameter	Value
Density Factor, η_d	0.08
Bytes per object, n_B	84.0
Constant, a	6.0

The effect of coupled transmission power and message content control scheme gives us an optimum message length that can maintain a less congested communication channel without lowering the map accuracy. The results of this approach have been described in the experimental results section.

Chapter 4: Experimental Results and Discussions

4.1 Environment Reconstruction and Understanding

As discussed in previous chapters, this experiment is divided in two parts. First part deals with the vehicle environment reconstruction and understanding using the fusion of camera and LIDAR data. This process is done in several steps: alignment of multiple point cloud sets for environment reconstruction, irregularity detection, pedestrian and vehicle detection. Experimental results for all these steps have been illustrated in this chapter.

4.1.1 EM-ICP based Environment Reconstruction

In the first step, we report the environment reconstruction results of the proposed EM-ICP for a variety of point clouds – urban vs. rural (refer to Figure 4-1) and road vs. intersection (refer to Figure 4-2). In each case, we have aligned and combined ten point cloud sets (each set contains around 200,000 points). It can be observed that the density of the point cloud increases along the moving direction of HV. This is consistent with the fact that more overlaps exist in the point clouds in the front than behind the vehicle. We have also noticed that the reconstruction of urban environment appears to be more accurate than that of rural one. One possible explanation is that man-made structures in urban environments (e.g., buildings, traffic signs etc.) better fit the proposed EM-ICP algorithm than those natural objects such as trees in rural environments. Similar findings can be

found in the previous work on efficient variants of ICP algorithms [63]. To qualitatively evaluate the accuracy of our EM-ICP algorithm, we have calculated the x-component of the translation of the HV in each frame and compared it with the ground-truth (GT) data supplied by KITTI database [54]. The comparison result has been shown in Figure 4-3.

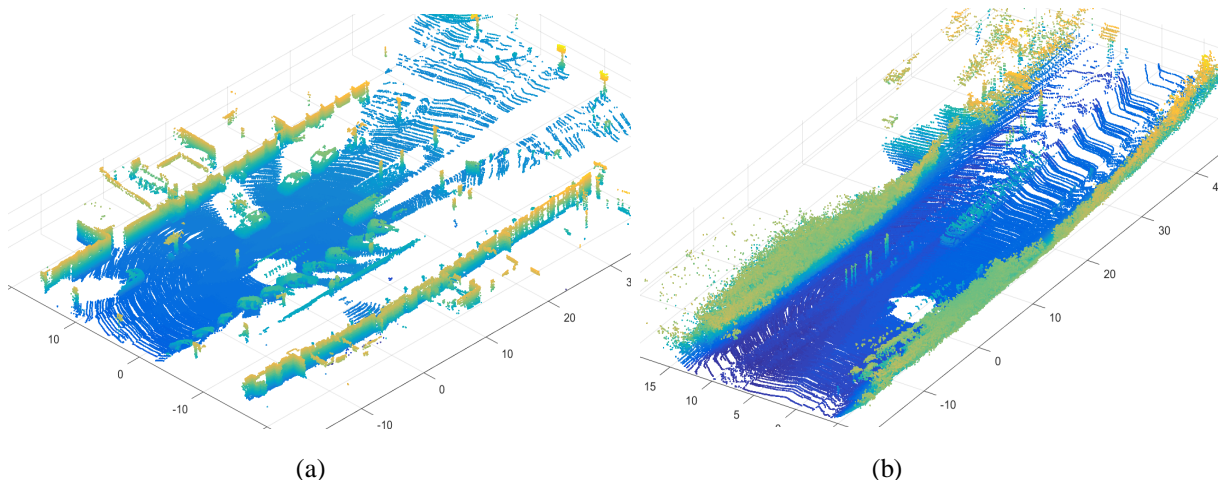


Figure 4-1: Reconstructed vehicle environments from a collection of 10 point clouds- a) urban vs. b) rural

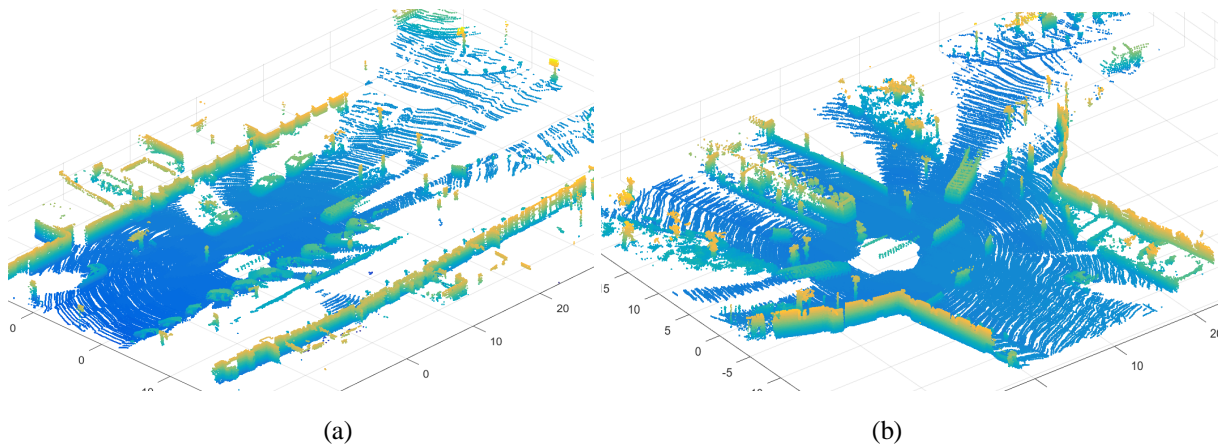


Figure 4-2: Reconstructed vehicle environments from a collection of 10 point clouds- a) road vs. b) intersection

After the alignment process, foreground (moving objects) is separated from background (stiff objects) for further analysis (e.g. irregularity detection).

4.1.2 Motion Trajectory Analysis and Irregularity Detection

In this experiment, we report our analysis with the extracted foreground (moving objects) and background (still objects). The objective of this analysis is to propose a technique to predict irregular movements of the objects. Moving objects will raise a warning if it has the potential of colliding with the HV in the near future (i.e., an irregular event). An exemplar result of the proposed irregularity detection method is shown in Figure 2-10, where the movement of a cyclist is detected as irregular. Some more results have been shown in Figure 4-4, where three different scenarios have been illustrated; (a) pedestrians crossing the road, (b) a cyclist, and (c) a moving car. The moving objects are shown in red color, and the blue dots are the location of predicted collision with HV if the trajectories are not changed. Although, these scenarios are detected as critical, there is no actual data of collision as these are some real-life data and the objects changed their trajectories before any kind of collision. Hence, we could not compare our predicted results with any ground truth.

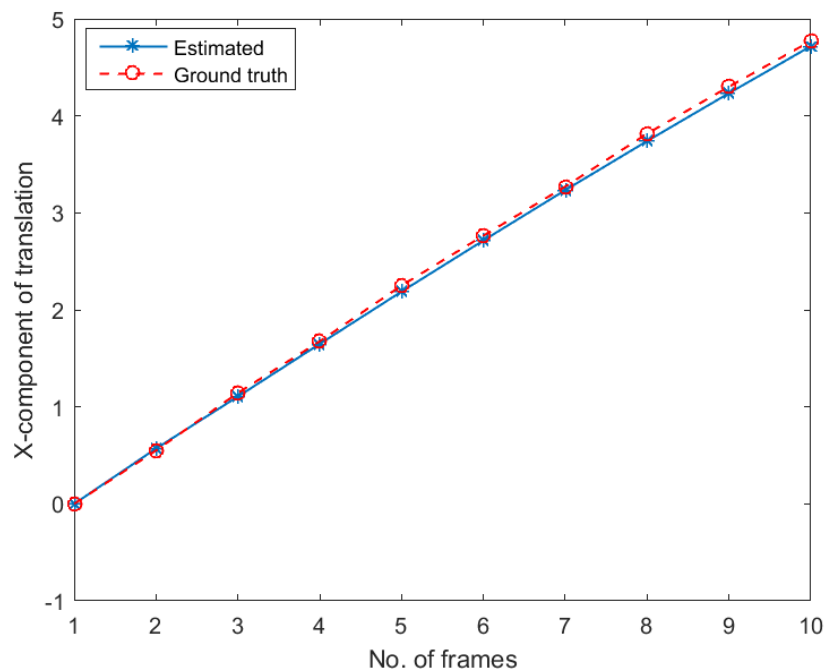
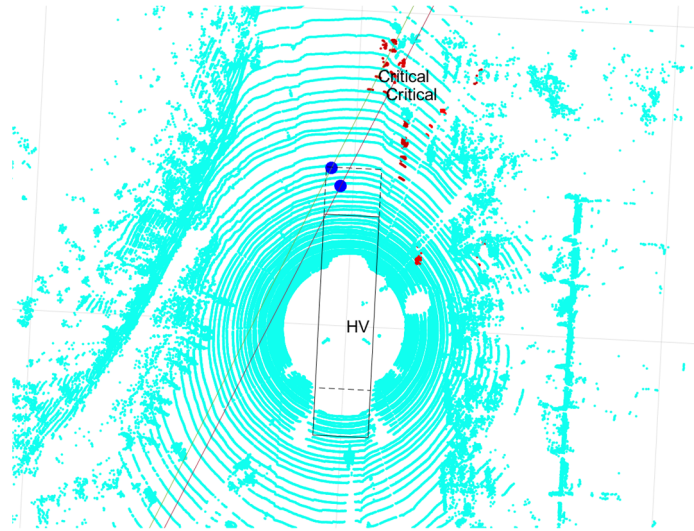
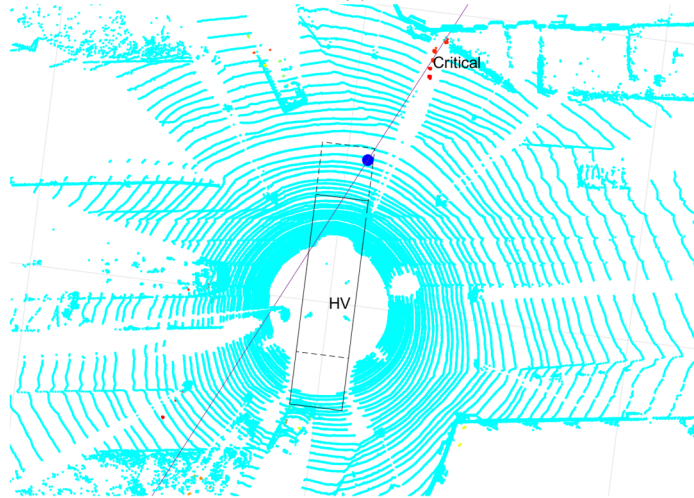


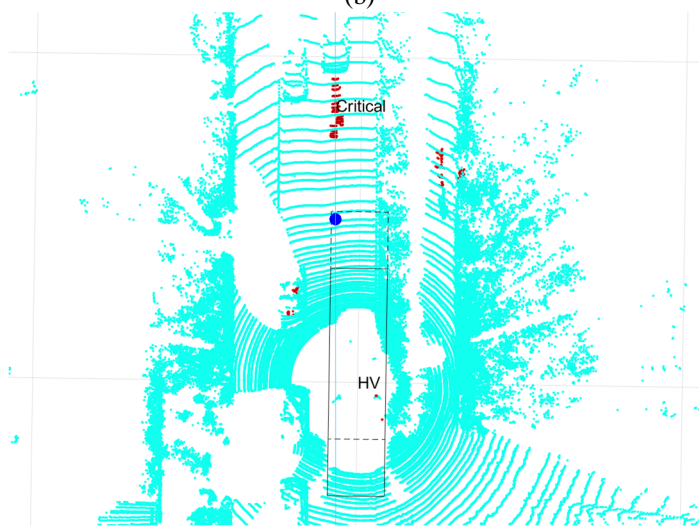
Figure 4-3: Alignment accuracy comparison (ground-truth is supplied by KITTI [54])



(a)



(b)



(c)

Figure 4-4: Irregularity detection in different scenario (a) pedestrian, (b) cyclist, (c) vehicle

4.1.3 Pedestrian Detection Results

Histogram of oriented gradient (HOG) features extracted from distance maps of the point cloud clusters are used to train a linear SVM which is later used to detect pedestrian in test point clouds. For the training purpose, total 106 samples have been used (positive 51, negative 55). All the samples are the D-maps of the point clouds of pedestrians (positive) and background (negative) objects. For the testing purpose, some point cloud sequences from KITTI dataset are applied. All the sequences have single or multiple moving pedestrians, and complex background (i.e. multiple non-human objects like trees, cycles, roadside walls etc.). A sample result has been shown in Figure 2-13. Some more results are illustrated in Figure 4-5, where detected pedestrians are shown in red color.

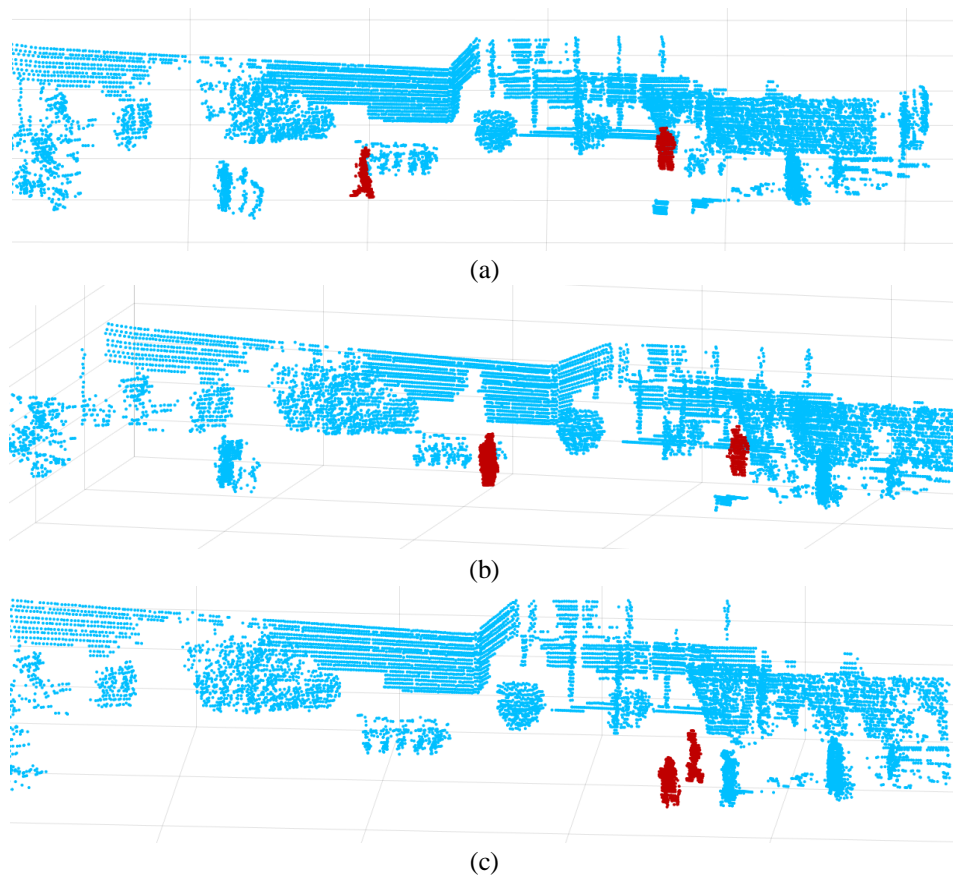


Figure 4-5: Results of HOG-SVM based pedestrian detection

We have tested the proposed pedestrian detection technique on five scenarios of the KITTI dataset (category: raw data). From each scenario, 25 consecutive point cloud frames have been taken (total 125 frames). For evaluation, ground truth (GT) has been set manually, i.e. an object is manually labeled as pedestrian or not. Overall and frame-wise successful detection rates are reported. Number of background objects wrongly detected as pedestrians are also recorded. The overall result is shown in Table 3-1. Here the sequences are continuous point cloud sets. So, same person can be found in multiple consecutive frames. But, same person in different frames are considered as different pedestrians, because all the frames are processed independently. For example, in the first sequence, all 25 frames contain same person but in different locations. As a result, the total number of pedestrians in 25 frames of sequence 1 is calculated as 25, not 1.

Table 4-1: Overall and frame-wise pedestrian detection accuracy

Sequence	Number of Frames	Number of Pedestrians (GT)	Detection Accuracy (%)	Wrongly Detected (objects/frame)
1	25	25	84.0	0.48
2	25	100	71.0	0.52
3	25	61	81.97	0.44
4	25	50	78.0	0.56
5	25	25	100.0	0.32
Total	125	261	78.93	0.46

4.1.4 Vehicle Detection Results

Like the pedestrian detection, HOG based features are also used to detect vehicle. But, for vehicles position and intensity-included HOG (PIHOG) has been used instead of traditional HOG features. Also, instead of using only point cloud information, we have used the six-dimensional environment map that is found after the fusion of RGB image and XYZ point cloud information. Later front view

of this map has been projected to a 2D RGB image to apply PIHOG. The main advantage of this image over regular RGB image (from camera) is that it solves the scaling problem of the regular RGB image, as in the projection procedure actual Z-values of the points are used.

For the training purpose, we have used 48x48 images of vehicles (positive) and background (negative) objects cropped from the 2D projection of front view of the map. Total 215 images (102 positive, 113 negative) have been used in training phase. PIHOG has been applied to these positive and negative images to extract features. Similar to pedestrian detection, these features are used to train a linear support vector machine (SVM). Then, this trained model has been tested using five different scenarios (raw data – city and road environments) from the KITTI dataset. The location of the vehicles in these scenarios are labelled manually for evaluation. To extract the features, a 40x40 sliding window (with 50% overlapping) is used for each frame. This window is reshaped to 48x48 and PIHOG features are extracted from each window.

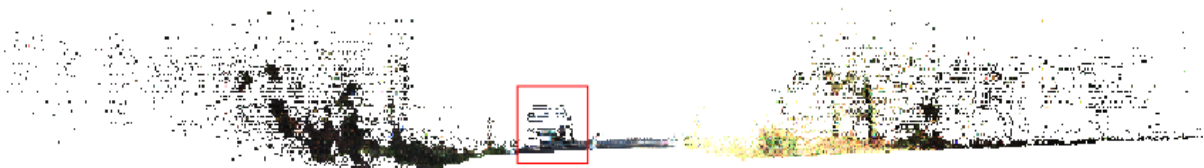
One of the evaluation results has been shown in Figure 2-17. More results have been shown in Figure 4-6, where vehicles have been detected (shown in red boxes) from multiple scenarios. The results on test data are shown in Table 4-2, where both overall and frame-wise vehicle detection results are recorded. Actual number and location of the vehicles (ground truth) are labelled manually for the comparison. 25 frames have been used for each sequence (i.e. total 125 frames for all 5 sequences). The number of successful and wrong detection is recorded for each frame. Detection accuracy is shown in percentage, and the wrong detection is expressed in per frame. Here we can see that the overall accuracy is around 73%. The reason behind the low accuracy is that we have only used the front view of the map, which can miss some vehicle because of occlusion. If other views are considered this accuracy can be improved.



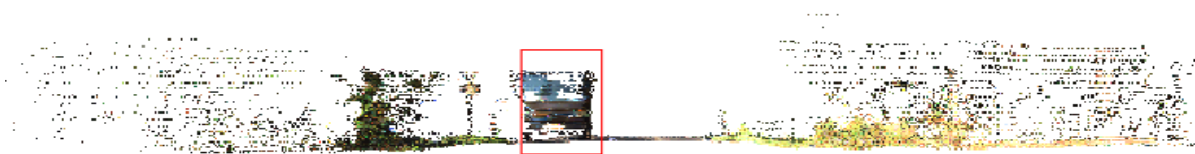
(a)



(b)



(c)



(d)

Figure 4-6: Results of PIHOG-SVM based vehicle detection

Table 4-2: Overall and frame-wise vehicle detection accuracy

Sequence	Number of Frames	Number of Vehicles (GT)	Detection Accuracy (%)	Wrongly Detected (objects/frame)
1	25	109	66.06	0.04
2	25	26	76.92	0.04
3	25	30	83.33	0
4	25	26	80.77	0.8
5	25	60	76.67	0.4
Total	125	251	73.31	0.256

4.2 Performance of Power Control Algorithm

The outcome of the proposed method (power control coupled with content control) is compared to other two schemes: 1) No content control, no power control (Baseline), and 2) Distance based content control (DBC), but no power control. Both these schemes are simulated in NS-3 using 10dBm transmission power. Vehicle trajectories of a 4-kilometer long highway detailed in [22] is used in NS-3 simulation. Three log-distance propagation path loss [61] (with the path-loss exponents of $\alpha_1 = 1.9$ and $\alpha_2 = \alpha_3 = 3.8$, and reference distances of $d_0 = 1$, $d_1 = 200$, and $d_2 = 500$) model is assumed for the wireless communication channels in 5.9 GHz DSRC band. For channel fading, default ns-3 Nakagami fading model is used with distance-dependent parameter m ; for distance smaller than 80 meters, $m=1.5$; otherwise, $m=0.75$. CSMA/CA with enhanced distributed channel access (EDCA) is used as MAC protocol.

Figure 4-7 illustrates the comparison results among the proposed algorithm and other two schemes for two different vehicle densities, (a) 125vehicles/Km and (b) 250 vehicles/Km. The results show that the proposed scheme outperforms the baseline in both environments. DBC still gives better result for the close distance (less than 225m), but the performance of the (power + content) control algorithm is very close. For the distance greater than 225m, combined effect of power and content control gives better accuracy than using DBC alone. The reason behind this is that in DBC, closer objects get higher priority during inclusion process. In case of channel congestion, it updates information of only the neighboring vehicles, not the distant ones, to fit the reduced message size. But in the proposed algorithm, transmission power is adjusted to keep the message size greater than a certain level, and thus the information of the distant vehicles is also updated.

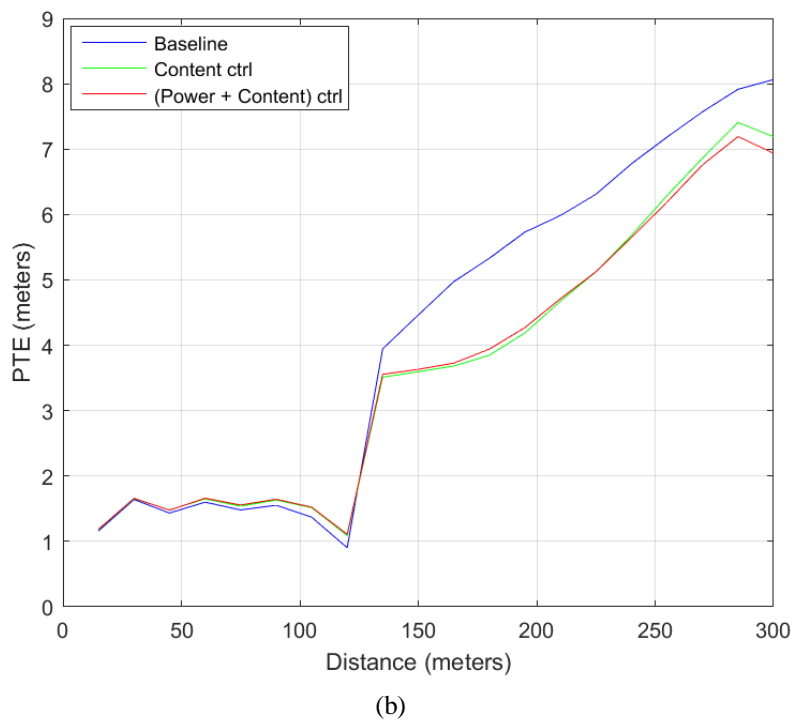
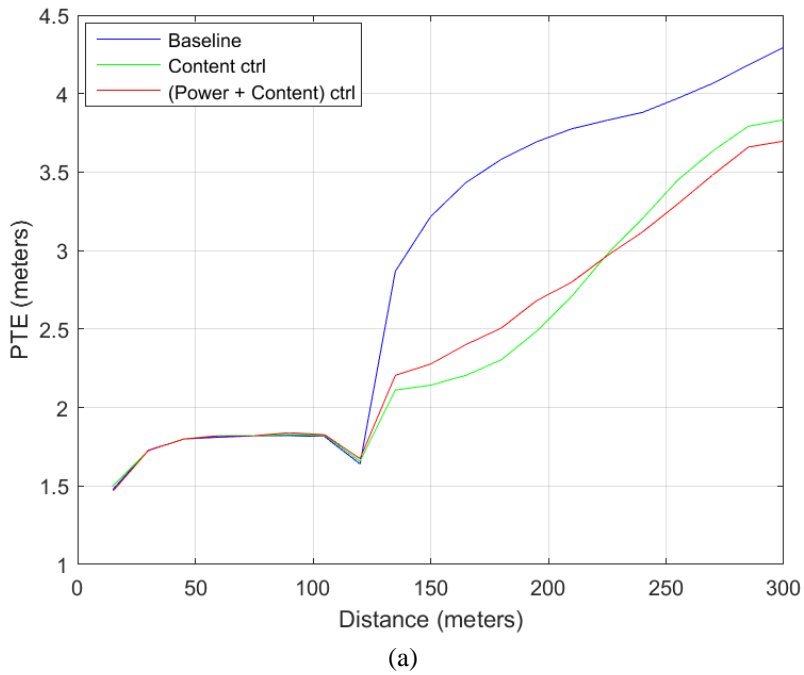


Figure 4-7: PTE comparison among three schemes for vehicle densities (a) 125 vehicles/Km, and (b) 250 vehicles/Km

Chapter 5: Conclusion and Future Works

In this work, we have presented a fusion technique that uses both image and point cloud data to reconstruct the vehicle environment in autonomous vehicle system. We have proposed a robust EM-ICP algorithm for reconstructing the vehicle environment from a collection of point clouds. Later, RGB color information has been combined with it for better representation of the environment that gives a six-dimensional map (RGB+XYZ) of the neighborhood. Moving objects including vehicles, pedestrians, and cyclists are detected and tracked through consecutive frames. Based on the analysis of moving trajectories, we propose to address the issue of irregularity detection under the context of intelligent vehicles and attempt to predict irregular events. Our preliminary experimental results have shown successful reconstruction of vehicle environments for a variety of situations (urban and rural, road and intersection). We also exploited HOG based object detection techniques to detect pedestrians and vehicles. To detect pedestrians, two-dimensional distance-maps of the clusters in a point cloud set are calculated and HOG is applied to this map. For vehicle detection, we have used the fused map (six-dimensional), and a modified HOG (PIHOG) has been applied to the front view (projected in a 2D image) of this map for feature extraction. In both cases, SVM with linear kernel is used for evaluation.

As the local sensors (camera, LIDAR) have range limitation, we have also discussed about a V2V communication framework that depends on the exchange of map information to increase the visibility of a vehicle. Sometimes, this framework can face the scalability problem because of high channel occupancy. To address this issue, performance of a content control strategy has been analyzed under different transmission power conditions. Based on these results a combination of transmission power and content control schemes has been proposed in this work. This algorithm

follows the distance based content control method to deal with the channel congestion, and at the same time adjusts the transmission power to maintain an optimum message length to attain an accurate map of the neighborhood. The accuracy is shown in the experimental result in terms of position tracking error (PTE).

Along this line of research, possible direction for next steps can be listed as follows:

- For irregularity detection, we could not measure the accuracy because of unavailability of actual irregular data. In future, we need to generate irregular scenarios both in lab and on-road environments to conduct more experiment with the proposed algorithm.
- In most of the point cloud sets, it is difficult to detect the objects that are in long distance from the HV, as points from distant objects are scattered in the point cloud. As a future work, we can develop an interpolation algorithm for point cloud using the fusion of LIDAR and image data.
- In both pedestrian and vehicle detection techniques, only front view (distance of the points for pedestrian, and projected front view for vehicle) of the environment has been used. For better accuracy, we also need to explore other views.
- Model based approaches have been used in vehicle and pedestrian detection techniques. In recent years, deep learning based approaches have gained popularity in recognition problems. We have plans to exploit some deep learning algorithms in vehicle environment understanding (e.g. vehicle or pedestrian detection, different types of vehicle classifications etc.).
- Beside pedestrians and vehicles, we need to develop detection techniques for other objects on road (e.g. traffic sign, road sign, trees, road divider etc.) for complete understanding of the environment.

- In this work, we have studied the fusion of local sensors, and V2V communication system separately. As a future works, we need to set up a combined experiment that will generate a complete map of the environment using local sensors and communicate this map with other neighboring vehicles.

References

- [1] J. H. Joung, K. H. An, J. W. Kang, M. J. Chung and W. Yu, "3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009.
- [2] S. Sivaraman and M. M. Trivedi, "A review of recent developments in vision-based vehicle detection," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Australia, 2013.
- [3] D. Geronimo, A. M. Lopez, A. D. Sappa and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239 - 1258, 2010.
- [4] L.-W. Tsai, J.-W. Hsieh and K.-C. Fan, "Vehicle Detection Using Normalized Color and Edge Map," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 850 - 864, 2007.
- [5] D. Guo, T. Fraichard, M. Xie and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA, 2000.
- [6] M. Bertozzi and A. Broggi, "Vision-based vehicle guidance," *Computer*, vol. 30, no. 7, pp. 49-55, 1997.
- [7] M. Enzweiler and D. M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179 - 2195, 2009.
- [8] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.
- [9] H. Men, B. Gebre and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011.
- [10] S. Granger and X. Pennec, "Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration," in *European Conference on Computer Vision (ECCV 2002)*, Berlin Heidelberg, 2002.
- [11] J. Hernandez and B. Marcotegui, "Point cloud segmentation towards urban ground modeling," in *Joint Urban Remote Sensing Event*, Shanghai, China, 2009.
- [12] M. Kusenbach, M. Himmelsbach and H.-J. Wuensche, "A new geometric 3D LiDAR feature for model creation and classification of moving objects," in *IEEE Intelligent Vehicles*

Symposium (IV), Gothenburg, Sweden, 2016.

- [13] B. Völz, H. Mielenz, R. Siegwart and J. Nieto, "Predicting pedestrian crossing using Quantile Regression forests," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016.
- [14] Navarro-Serment, L. E., C. Mertz and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1516-1528, 2010.
- [15] El-Halawany, A. M. Sherif, D. D. Lichti and N. El-Sheimy, "Detection of road curb from mobile terrestrial laser scanner point cloud," in *Proceedings of the ISPRS Workshop on Laserscanning*, Calgary, Canada, 2011.
- [16] C. Fernández, D. F. Llorca, C. Stiller and M. A. Sotelo, "Curvature-based curb detection method in urban environments using stereo and laser," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, 2015.
- [17] A. Y. Hata, F. S. Osorio and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, 2014.
- [18] C. Premebida, J. Carreira, J. Batista and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 2014.
- [19] F. Zhang, D. Clarke and A. Knoll, "Vehicle detection based on LiDAR and camera fusion," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014.
- [20] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162 - 1182, 2011.
- [21] Y. P. Fallah, C.-L. Huang, R. Sengupta and H. Krishnan, "Analysis of Information Dissemination in Vehicular Ad-Hoc Networks With Application to Cooperative Vehicle Safety Systems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 233 - 247, 2011.
- [22] G. Bansal, H. Lu, J. B. Kenney and C. Poellabauer, "EMBARC: error model based adaptive rate control for vehicle-to-vehicle communications," in *In Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications (VANET '13)*, New York, NY, USA, 2013.
- [23] M. Torrent-Moreno, J. Mittag, P. Santi and H. Hartenstein, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3684 - 3703, 2009.

- [24] G. Bansal, J. B. Kenney and C. E. Rohrs, "LIMERIC: A Linear Adaptive Message Rate Algorithm for DSRC Congestion Control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182 - 4197, 2013.
- [25] C.-L. Huang, Y. P. Fallah, R. Sengupta and H. Krishnan, "Adaptive intervehicle communication control for cooperative safety systems," *IEEE Network*, vol. 24, no. 1, pp. 6 - 13, 2010.
- [26] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for Vehicle Safety Communications," in *2011 IEEE Vehicular Networking Conference (VNC)*, Amsterdam, Netherlands, 2011.
- [27] M. Fanaei, A. Tahmasbi-Sarvestani, Y. P. Fallah, G. Bansal, M. C. Valenti and J. B. Kenney, "Adaptive Content Control for Communication amongst Cooperative Automated Vehicles," in *IEEE 6th International Symposium on Wireless Vehicular Communications*, Vancouver, BC, 2014.
- [28] P. Jenke, M. Wand, M. Bokeloh, A. Schilling and W. Straßer, "Bayesian point cloud reconstruction," *Computer Graphics Forum*, vol. 25, no. 3, pp. 379-388, 2006.
- [29] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.
- [30] M. Vlamincx, H. Q. Luong, W. Goeman, P. Veelaert and W. Philips, "Towards online mobile mapping using inhomogeneous lidar data," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016.
- [31] R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009.
- [32] F. Moosmann, O. Pink and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, China, 2009.
- [33] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, 2005.
- [34] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, p. 91–110, 2004.
- [35] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, vol. 38, no. 1, p. 15–33, 2000.
- [36] P. Viola, M. J. Jones and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice,

France, 2003.

- [37] M. J. Jones and D. Snow, "Pedestrian detection using boosted features over many frames," in *2008 19th International Conference on Pattern Recognition*, Tampa, FL, USA, 2008.
- [38] D. M. Gavrila, J. Giebel and S. Munder, "Vision-based pedestrian detection: the PROTECTOR system," in *2004 IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004.
- [39] D. M. G. a. S. Munder, "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle," *Int. J. Comput. Vision*, vol. 73, no. 1, pp. 41-59, 2007.
- [40] S. Munder and D. M. Gavrila, "An Experimental Study on Pedestrian Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1863 - 1868, 2006.
- [41] L. Zhao and C. E. Thorpe, "Stereo- and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148 - 154, 2000.
- [42] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito and J. Miura, "Pedestrian recognition using high-definition LIDAR," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011.
- [43] J. Behley, V. Steinhage and A. B. Cremers, "Laser-based segment classification using a mixture of bag-of-words," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.
- [44] F. Hegger, N. Hochgeschwender, G. K. Kraetzschmar and P. G. Ploeger, "People Detection in 3d Point Clouds Using Local Surface Normals," in *RoboCup 2012: Robot Soccer World Cup XVI*, Springer, Berlin, Heidelberg, 2013.
- [45] L. Spinello, K. O. Arras, R. Triebel and R. Siegwart, "A layered approach to people detection in 3D range data," in *In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*, Atlanta, Georgia, 2010.
- [46] S. Sivaraman and M. M. Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773 - 1795, 2013.
- [47] J. Kim, J. Baek and E. Kim, "A Novel On-Road Vehicle Detection Method Using π HOG," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3414 - 3429, 2015.
- [48] F. Erbs, A. Barth and U. Franke, "Moving vehicle detection by optimal segmentation of the Dynamic Stixel World," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011.
- [49] B. Li, T. Zhang and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional

- Network," *arXiv preprint arXiv:1608.07916*, 2016.
- [50] J. Eum, M. Bae, J. Jeon, H. Lee, S. Oh and M. Lee, "Vehicle detection from airborne LiDAR point clouds based on a decision tree algorithm with horizontal and vertical features," *Remote Sensing Letters*, vol. 8, no. 5, pp. 409-418, 2017.
- [51] A. Weinfield, J. B. Kenney and G. Bansal, "An Adaptive DSRC Message Transmission Rate Control Algorithm," in *18th ITS World Congress*, Orlando, Florida, 2011.
- [52] S. M. O. Gani, A. Tahmasbi, Y. P. Fallah, G. Bansal and T. Shimizu, "Evaluating performance of content and rate control schemes for scalable automated vehicles communication," in *23rd ITS World Congress*, Melbourne, Australia, 2016.
- [53] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, 1999.
- [54] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1237, 2013.
- [55] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189-206, 2013.
- [56] D. Chetverikov, D. Stepanov and P. Krsek, "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, p. 299-309, 2005.
- [57] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, Oregon, 1996.
- [58] B. Völz, H. Mielenz, R. Siegwart and J. Nieto, "Predicting pedestrian crossing using Quantile Regression forests," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, 2016.
- [59] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273 - 297, 1995.
- [60] J. Han, M. Kamber and J. Pei, *Data mining: concepts and techniques*, Elsevier, 2011.
- [61] C. Sommer and F. Dressler, "Using the Right Two-Ray Model? A Measurement-based Evaluation of PHY Models in VANETs," in *Proc. Of ACM International Conference on Mobile Computing and Networking (MobiCom)*, Las Vegas, NV, 2011.
- [62] Y. P. Fallah, C. Huang, R. Sengupta and H. Krishnan, "Congestion Control Based on Channel Occupancy in Vehicular Broadcast Networks," in *2010 IEEE 72nd Vehicular Technology*

Conference Fall (VTC 2010-Fall), Ottawa, ON, Canada, 2010.

- [63] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Quebec, Canada, 2001.