Master's Thesis

Title

A Study on Extraction and Integration of Sensor Information from Multiple LiDARs for Informative Dynamic Map

Supervisor

Associate Professor Shin'ichi Arakawa

Author

Shun Nawa

February 2nd, 2023

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

A Study on Extraction and Integration of Sensor Information from Multiple LiDARs for Informative Dynamic Map

Shun Nawa

Abstract

In recent years, attention has focused on technologies that use cameras, LiDAR sensors, and other devices mounted on vehicles to recognize the vehicle's surrounding environment for driving assistance and automatic driving. Each vehicle's onboard sensors alone can only provide information several hundred meters away and cannot detect objects in blind spots caused by other vehicles or buildings. Therefore, it is difficult for vehicles to accurately drive on the roadway by estimating their own position and accelerating/decelerating in advance. Therefore, research and development of dynamic maps is underway to integrate and share sensor information from each vehicle and roadside units to enable each vehicle to accurately grasp the surrounding environment in real time. A dynamic map is a digital map that combines dynamic information such as cars and pedestrians with 3D map information such as lane information and structures. Existing methods that integrate each in-vehicle sensor data to create dynamic maps do not obtain information such as the speed and direction of moving objects included in the integrated data, and thus cannot be applied to collision detection and course determination. It is also necessary to obtain information on objects that may start moving even if they are not currently moving, such as parked vehicles, as information that should be noted.

In this study, moving objects and potentially moving objects on the road are extracted, integrated, and represented on the same map. In this study, information that includes moving objects is called dynamic information, while parked objects that may move, such as stopped vehicles, are called semi-dynamic information. Static information, which is information such as roads and structures on roads is considered to be information that can be received from a third party in the future and is used in this study, based on the fact that there are companies and organizations that are working on 3D mapping. Global coordinates of sensor data are obtained by matching sensor data with a static map, which is a map consisting of only static information. Semi-dynamic information is extracted by taking the difference between the static map and sensor data. Dynamic information is obtained by detecting the dynamic part of the sensor data using existing methods and analyzing the point cloud of the detected part to calculate the speed and direction of travel. Finally, the extracted information is integrated on the static map.

Keywords

LiDAR
Dynamic Map
Dynamic Information
Semi-Dynamic Information
Point Cloud Registration
MEC (Multi-access Edge Computing)

Contents

1 Introduction **6**

2 Systems of Informative Dynamic Map **9**

2.1 Use Case and Prerequisite 9

2.2 System Model 9

2.3 Sensor Information Types 9

2.3.1 Static Information 9

2.3.2 Semi-Dynamic Information 10

2.3.3 Dynamic Information 10

2.4 A Concept of Informative Dynamic Map 11

3 A Method for Information Extraction and Integration from Multiple Sensors **12**

3.1 Overview 12

3.2 Coordinate Transformation 12

3.2.1 A Feature Used for Rough Alignment 13

3.2.2 A Method for Rough Alignment 13

3.2.3 A method for Fine-Tuning the Rough Alignment 14

3.3 Dynamic Information Extraction 15

3.3.1 Lidar-based Moving Object Segmentation (LiDAR-MOS) 15

3.3.2 Density-based spatial clustering of applications with noise (DBSCAN) 15

3.4 Semi-Dynamic Information Extraction 16

3.5 Information Processing 16

3.6 Integration 17

4 Evaluation of Proposed Method **18**

4.1 Data Collection and Preparing 18

4.1.1 Data Collection 18

4.1.2 Preparing Multiple Sensor Data and Static Map 18

4.2 Visualization of Extracted Information 19

4.2.1 Visualization of Dynamic Information 19
4.2.2 Visualization of Semi-Dynamic Information 19
4.3 Accuracy of Extraction and Integration 19
4.3.1 Accuracy of Registration 19
4.3.2 Dynamic Information Extraction 20
4.3.3 Semi-Dynamic Information Extraction 21
4.3.4 Velocity Vectors of Moving Objects 21
4.4 Limitations and Further Direction 21

5 Conclusion **31**

Acknowledgments **32**

References **33**

List of Figures

1 System model 10
2 Overview of proposed method 12
3 Point registration 15
4 A photograph taken from the position of the sensor 18
5 Visualization of a extracted moving car information 23
6 Visualization of a extracted moving bicycle information 24
7 Visualization of a extracted pedestrian information 25
8 Visualization of a extracted parked bicycle information 26
9 The case when a moving object is not detected 27
10 The case when a parked bicycle is not detected 28
11 The case when a parked bicycle is not detected 29
12 Velocity vectors of moving objects 30

1 Introduction

In recent years, attention has focused on technologies that enhance traffic safety by installing cameras and LiDAR sensors in vehicles to monitor the surrounding environment. For example, cars equipped with pre-crash brakes that use cameras and millimeter wave radar to measure the distance between the vehicle and objects in front are already commercially available, as are collision warning functions that alert the driver of a potential collision. However, the information that can be obtained by in-vehicle sensors is limited to the area that can be seen from the vehicle, so objects in blind spots by other vehicles or buildings, or objects outside the detection range of the sensors, cannot be recognized [1]. Therefore, a method to supplement sensor information by sharing information through V2X (Vehicle-to-Everything) communication between vehicles or between vehicles and roadside units was proposed [2–5]. In recent years, dynamic maps, which enable more efficient use of traffic environment information by linking sensor information, road traffic information, and 3D map information obtained through V2X communication, have been attracting attention. 3D map information is a map that records road signs and buildings in three dimensions in addition to the information contained in conventional two-dimensional maps. Dynamic maps include a wide range of traffic environment information and can be used not only for applications that require a wide range of information, such as traffic jam forecasting, but also for applications that use a relatively narrow range of information, such as collision detection.

In order to expand the sensing range of each vehicle and improve sensing accuracy, there is an effort to integrate the objects obtained by each onboard sensor into a single map using a centralized server [6]. In Ref. [6], the local map created by each vehicle is integrated with the global map using GPS coordinates in Centralized Server. However, information such as the speed and direction of moving objects in the map is not obtained. Information on the position, speed, and direction of dynamic objects is required at a minimum for collision detection and route determination [7]. It is also necessary to obtain information on objects that may start moving even if they are not currently moving, such as a parked car, as information that should be noted. Therefore, there is a method for detecting dynamic objects, which is called [8]. Reference [8] is a method for detecting moving objects using a range image created using point cloud data obtained by a self-vehicle and a residual image between consecutive range images over time. It is possible to detect moving objects regardless of their size or shape. Since the range image is an image that contains distance data from one point to each of the surrounding point clouds when viewed from a single point, it is necessary to use data from a single LiDAR sensor. Also, since objects that are stopped are not detected, information on cars and pedestrians included in dynamic object information cannot be detected while they are stopped.

This study aims to create maps that can be applied to collision detection and course determination. Therefore, this study will extract moving objects and potentially moving objects on the road. Compared to static objects, dynamic objects have a higher risk of collision with vehicles and people, and it is more important to be able to extract objects regardless of size and shape than to identify them. Therefore, this study does not perform object identification, but rather detection and extraction of dynamic objects. In addition, objects that are not moving but may move, such as a parked vehicle, are also extracted without identification because the risk of collision is higher than static objects. To obtain the global coordinates of the extracted information, the sensor information is integrated on the same map. After integration, information on the position, direction of movement, and speed of the extracted object is calculated and displayed to represent the real-world environmental information.

In this study, maps that do not include map information such as lane information and geographic information, but only three-dimensional static information such as buildings and roads are called static maps, and dynamic and semi-dynamic information is extracted from multiple LiDAR sensors and integrated with the static map. Reference [8] is used as a detection method for the dynamic part of the point cloud data. The method of Reference [8] requires the use of data from a single LiDAR sensor. Therefore, dynamic information is extracted before integrating the data from each LiDAR sensor. The semi-dynamic information is extracted from the difference between the static map and each LiDAR sensor data. Since related work [6] has been conducted on matching in-vehicle sensor data with the global map using GPS, it is assumed that global matching is possible with a certain degree of accuracy. Therefore, in this study, we use general point cloud registration for local positioning and integration of a limited range of the static map and sensor data. The center of gravity is obtained from the extracted point cloud of dynamic information, and information on the position, direction of motion, and velocity of the dynamic object is obtained by calculating the shift of the center of gravity between each frame.

2 Systems of Informative Dynamic Map

2.1 Use Case and Prerequisite

The maps created in this study are positioned as a map that may be used not only by vehicles, but also by pedestrians, logistics providers, and all other users of transportation, including individuals and groups. For example, it can be applied to collision detection applications because the map contains information on the coordinates, direction of travel, and speed of moving objects. The map also includes information on objects that may be moving, so it could be used in applications for course determination.

The premise of this study is that a static map, which is a map consisting only of static information, can be provided by a third party and is assumed to be owned in advance. In addition, based on related research [6], it is assumed that global positioning of the static map and sensor data can be achieved with a certain degree of accuracy using GPS.

2.2 System Model

Sensing of the surrounding environment is performed by multiple LiDAR sensors mounted on each vehicle and roadside units. The obtained sensor data is sent to the MEC with or without pre-processing. Since the created map is intended for use in applications that require real-time information, such as collision detection, the map must be created in real-time. Therefore, MEC is used because it is possible to update and share the created map with lower latency than using a centralized server. It is decided whether to do pre-processing or not based on which will reduce the overall time. In the MECs, dynamic and semi-dynamic information is extracted, and the extracted information is integrated with a static map using the method proposed in this study.

2.3 Sensor Information Types

2.3.1 Static Information

Information that has not changed for a long period of time, such as roads and structures on roads. Since some companies and organizations are already working on 3D mapping of static information, this study assumes that static information can be received from these

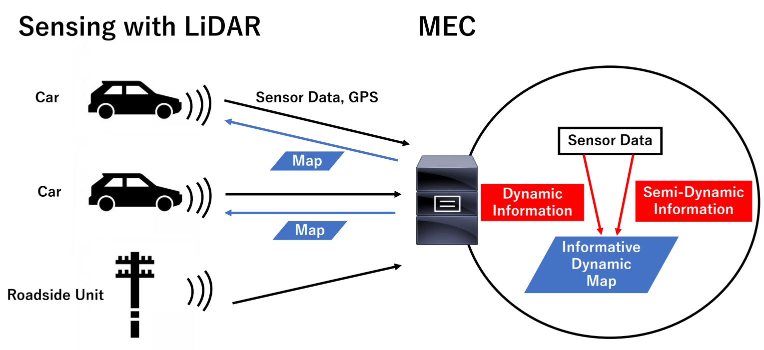


Figure 1: System model

third parties in the future, and we already have such information.

2.3.2 Semi-Dynamic Information

Information such as accident information, road construction information, and other information whose location, range, and occurrence time are irregular. In this study, objects that are actually on the road, such as parked vehicles and construction signs, are considered as objects to be acquired by sensors. In this study, parked bicycles are also considered as semi-dynamic information.

2.3.3 Dynamic Information

Information such as vehicles and pedestrians, where the position of the object is not fixed, but moves, or is fixed but only for a short period of time. Generally, dynamic information includes information such as traffic signal information, which has a short update cycle of attribute even if it is in a fixed position. However, traffic signal information is considered as information that can be received from traffic control centers. In this study, only information on objects whose positions are moving is included in dynamic information.

2.4 A Concept of Informative Dynamic Map

The map created in this study focus on dynamic and semi-dynamic information. Dynamic information is an important information for collision detection and course determination. Semi-dynamic information is also an information that needs attention because it is an object that may move. The method proposed in this study enables both types of the information to be obtained from multiple sensors and represented on the same map, making it possible to share the information widely. Dynamic and semi-dynamic information is processed on demand, and the calculated information is also displayed on the map.

The first feature of the Informative Dynamic Map created in this study is that it displays not only the position of objects in dynamic information, but also their direction and speed of movement. The second feature is that the map displays semi-dynamic information that may move, such as the position of a parked vehicle. This map can easily be used for collision detection and course determination applications because it shows the direction and speed of moving objects. By obtaining information on potentially moving objects, it will also be possible to determine a course in response to objects that suddenly start moving.

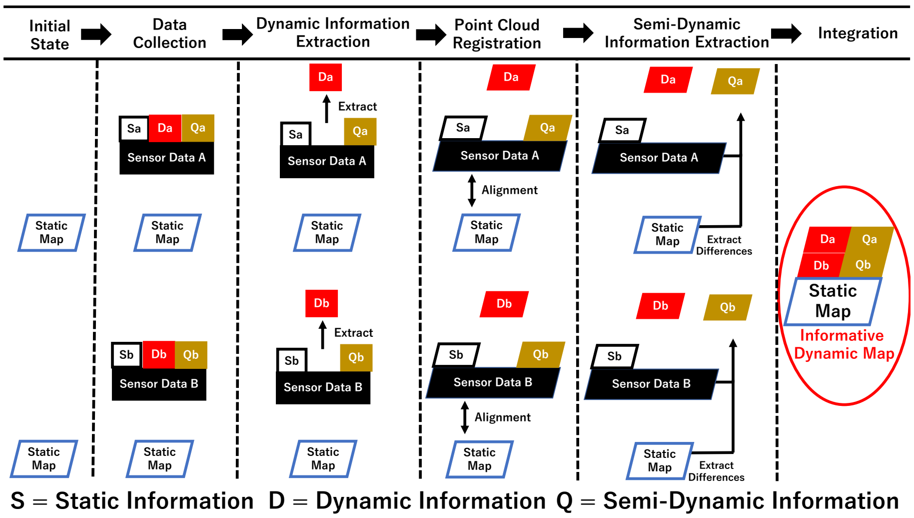


Figure 2: Overview of proposed method

3 A Method for Information Extraction and Integration from Multiple Sensors

3.1 Overview

This study was conducted to extract dynamic and semi-dynamic information from multiple LiDAR sensor data and to make a map by integrating the extracted information into the static map. The difference between the static map and sensor data is used to extract semi-dynamic information. For this purpose, the sensor data and the static map must be aligned. Dynamic information is extracted and removed from the sensor data prior to alignment because the accuracy of alignment is getting worse if dynamic information remains. After extracting dynamic and semi-dynamic information, they are analyzed, and the information and analysis results are integrated into the static map. The global coordinates of the sensor data obtained from this alignment are also necessary to integrate the extracted information on the static map.

3.2 Coordinate Transformation

To obtain the global coordinates of each sensor data, align the sensor data with the static map by point cloud registration. In this study, RANdom SAmple Consensus (RANSAC) estimation using Fast Point Feature Histograms (FPFH) features is used for rough positioning, and then Iterative Closest Point (ICP) is used for fine-tuning. Dynamic information can be extracted before point cloud registration. The more matching point clouds there are, the more accurate the alignment will be. Therefore, dynamic information is extracted and removed from sensor data before point cloud registration.

3.2.1 A Feature Used for Rough Alignment

In this study, FPFH is used as a feature for rough alignment. FPFH [9] is a feature that is a faster version of PFH [10] and is a histogram of the normal vector of the point of interest compared to the normal vector of the surrounding points.

First, estimate the normal vectors of an arbitrary point p_t and its neighbor point p_s are estimated. Next a set of values α, ϕ, θ between p_t and p_s are calculated as follows:

$$\begin{aligned} \alpha &= v \cdot n_s \\ \phi &= u \cdot \frac{p_s - p_t}{d} \\ \theta &= arctan(w \cdot n_s, u \cdot n_s) \end{aligned} \tag{1}$$

Note that we define $u = n_t, v = u \times \frac{p_s - p_t}{||p_s - p_t||_2}, w = u \times v$. α, ϕ, θ is represented by a histogram, and the histogram is called Simplified Point Feature Histogram (SPFH). When the SPFH values of an arbitrary point p_t is denoted by $h_{SPFH}(p_t)$, the FPFH values of the p_t are calculated as follows.

$$FPFH(p_t) = SPFH(p_t) + \frac{1}{k} \sum_{i=1}^{k} \frac{1}{\omega_i} \cdot SPFH(p_i) \tag{2}$$

ω_i represents a distance between p_t and its neighbor point p_i.

3.2.2 A Method for Rough Alignment

RANSAC [11] is used to estimate coordinates for rough alignment. Alignment is done by matching the FPFH features of the static map and sensor data. RANSAC is a method of learning parameters by excluding outliers from data containing outliers. The algorithm of RANSAC is outlined below:

1. Randomly select n points from data.

2. The provisional model C_1 is learned from the n points.

3. Calculate the error between the predictions calculated using C_1 and the actual data point for all data points.

4. The point whose error is smaller than the threshold is considered normal.

5. Using only normal points, repeat Step2 through Step4 a specified number of times to find the best performing model C_n.

6. Points which the error between the predictions calculated using the final model C_n and the actual data point is greater than a threshold are removed as outliers.

3.2.3 A method for Fine-Tuning the Rough Alignment

ICP is an algorithm used to align two 3D point clouds. ICP is used for fine-tuning of alignment. Formally, the process is as follows:

1. Search for corresponding points set $\kappa = (p, q)$ from target point cloud P and source point cloud Q using a nearest neighbor search algorithm.

2. Find transformation T such that the objective function $E(T)$ defined over the corresponding points set is minimized.

3. Transform the target point cloud by T.

4. Iterate Step1 to Step3

5. End the process if the difference between $E(T)$ at the kth and tth iterations is less than the threshold.

In this study, point-to-plane ICP algorithm [12] is used and KDTree is used as nearest neighbor search algorithm. The point-to-plane ICP algorithm uses an objective function below.

$$E(T) = \sum_{(p,q)\in\kappa} ((p - Tq) \cdot n_p)^2 \tag{3}$$

n_p is the normal of point p.

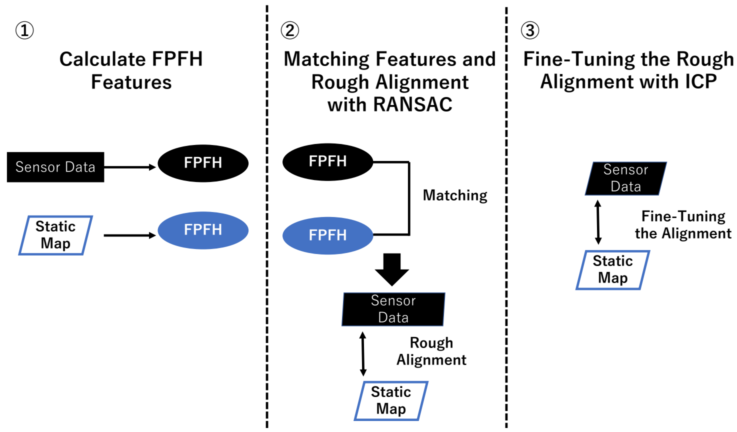


Figure 3: Point registration

3.3 Dynamic Information Extraction

Segment the moving objects in data and extract dynamic information from sensor data. In this study, LIDAR-MOS [8] is used for segmentation. After the segmentation, clustering the segmented part with DBSCAN to calculate the coordinate of moving objects and to remove false positive portion.

3.3.1 Lidar-based Moving Object Segmentation (LiDAR-MOS)

The moving object segmentation method using LiDAR-MOS is described below. First, point cloud data is converted into range images $\{R_i\}_{i=1}^{N}$. A range image is two-dimensional image containing distance information. Next, a residual image is generated. The residual image is generated from the difference between the range image R_t from the current scan and the range image R_{t-q} from the previous scan. In this study, only the residual image at $q = 1$ is used, but any number can be set to q. Finally, the R_t and the created residual images are used as input to the CNN (convolutional neural network) to segment the moving object in the range image and reconvert the segmented part to point cloud.

3.3.2 Density-based spatial clustering of applications with noise (DBSCAN)

DBSCAN [13] is a density-based clustering. Given a set of points in some space, points that are closely packed together are grouped together, and points in low-density regions are considered as outliers. The algorithm is outlined below:

1. Set two parameters eps and $minPts$ first.

2. Randomly select data point p from the data set, and set cluster label $C = 1$

3. Identify the data point ps included in the range of eps from data point p.

    (a) If the number of ps is less than $minPts$, label p as noise and go to Step4.

    (b) If the number of ps is more than or equal to $minPts$, label p as C.

4. Iterate Step2 until no points labeled C can be found in qs.

5. Update the cluster label $C+ = 1$ and randomly select a new data point p which is not labeled from the data set

6. Iterate Step2 to Step4 until all data points in the data set are labeled.

3.4 Semi-Dynamic Information Extraction

Semi-dynamic information is obtained by removing dynamic information from detected changes between the static map and sensor data after point cloud registration. This section describes the method of change detection. Finally, DBSCAN is used only to reduce noise.

octree_change_detection provided by python_pcl is used to change detection. python_pcl is a python binding to the Point Cloud Library (PCL). This code represents a point cloud as an octree and extracts the parts where the structure of the octree has changed. Octree is a data structure that has a tree with the entire space as the root node and recursively divided into eight parts down to the smallest voxel. Change detection by octree is faster than the simple method using distance, and the noise can be reduced by adjusting the resolution.

3.5 Information Processing

The extracted information is processed as needed. For example, dynamic information can be easily applied to collision detection applications by determining the speed and direction of travel of objects. For semi-dynamic information, it is possible to predict in which direction the objects are in danger of moving suddenly by determining the direction in which it is stopped.

3.6 Integration

The extracted dynamic and semi-dynamic information is added to the static map. Each information is added based on the global coordinates obtained from registration.

Figure 4: A photograph taken from the position of the sensor

4 Evaluation of Proposed Method

4.1 Data Collection and Preparing

4.1.1 Data Collection

In this study, Livox Horizon was used as sensor. Livox Horizon has a horizontal FOV of $81.7°$ and a vertical FOV of $25.1°$. The frame rate is 10 Hz. The number of the point clouds is $240,000$ per second. The reason for using LiDAR is that it can obtain information on the distance from the sensor to the object and its shape, which is necessary for applications such as collision detection. The data was collected on January 24, 2023, at the intersection in front of Seven Eleven (Poplar Street Welfare Center) in Osaka University. A photograph taken from the position of the sensor is shown in Figure 4. The position of the sensor is fixed.

4.1.2 Preparing Multiple Sensor Data and Static Map

In this study, the data obtained by one LiDAR sensor is divided and use them as data obtained by two different sensor data. The data is divided in a plane vertical to the ground with the direction the sensor is facing as the axis. A bicycle is added to the sensor data as semi-dynamic information.

Static map uses data at the moment when moving objects and the bicycle, which is semi-dynamic information, are not included.

In this study, LIDAR-MOS is used for moving objects segmentation. However, LiDAR-MOS is tuned using data from Velodyne HDL-64E (KITTI data set). Therefore, the KITTI data set is cropped to equal half of the Horizon viewing angle and the network is retrained.

4.2 Visualization of Extracted Information

4.2.1 Visualization of Dynamic Information

The extracted dynamic information (cars, bicycles, and pedestrians) is visualized as shown in the Figure 5, Figure 6, Figure 7. Object detection is succeeded to all three types of objects. The point cloud indicated by the red is the dynamic information, the white is the static map, and the yellow is the semi-dynamic information.

4.2.2 Visualization of Semi-Dynamic Information

The extracted semi-dynamic information is visualized as shown in the Figure8. In this study, a parked bicycle is extracted as semi-dynamic information.

4.3 Accuracy of Extraction and Integration

4.3.1 Accuracy of Registration

To evaluate the alignment, the alignment was conducted with an initial error. The initial error was set to 2 meters in the x, y, and z directions, respectively, and a 45° counterclockwise rotation around the z-axis. Since the sensor position is fixed, the alignment error is calculated from the gap of the sensor position. The average of the absolute values of the error between the original sensor position and the aligned sensor position in the x, y, and z directions is calculated and the result is in the table 1. Errors between consecutive frames have a significant impact on the calculation of direction of travel and velocity from dynamic information. Therefore, a moving distance average of the fixed sensor position after alignment between consecutive frames is calculated and the result is shown in the table 2. The error is roughly 10 cm at most for each direction, but since the frame rate is

Table 1: The average of absolute value of the alignment error

Sensor Data	$x(m)$	$y(m)$	$z(m)$
Left Side Data	0.025	0.019	0.18
Right Side Data	0.014	0.040	0.067

Table 2: The moving distance average of position of the fixed sensor between frames

Sensor Data	$x(m)$	$y(m)$	$z(m)$
Left Side Data	0.028	0.024	0.13
Right Side Data	0.022	0.034	0.10

10 Hz, the error is about 1 m/s. Since the speed of a pedestrian is about 1 m/s, the error might have a significant effect on calculation of direction the pedestrian is traveling. One way to reduce alignment errors is to use sensors that can obtain information over a wider area. This is because the more area information that can be obtained, the more points that can be matched with the static map and improve the accuracy of alignment. We will verify this using a 360° LiDAR sensor in the future.

4.3.2 Dynamic Information Extraction

There are moments when the object is not detected even though it is moving. In the case of the following Figure 9, the object was not recognized as a dynamic information, so it is considered as semi-dynamic information. A possible reason for the inaccurate detection of moving objects is that the type of LiDAR is different between the data used to train LiDAR-MOS and the data collected in this study. The accuracy would be improved if using Velodyne's sensors. Increasing the number of residual images used to train the LiDAR-MOS model would also improve accuracy.

There was also a moment when a part of semi-dynamic object, which is a parked bicycle, was detected as a moving object. This moment is shown in the figure 10. Semi-dynamic information was mistakenly detected as a moving object only when there was a moving object in the back. LiDAR-MOS detects moving objects by converting point clouds into a distance image, so it is expected that the areas that overlap with moving objects in the image are likely to be detected as moving objects incorrectly. False positives for semi-dynamic information could be reduced by analyzing semi-dynamic information in the time direction.

4.3.3 Semi-Dynamic Information Extraction

Semi-dynamic information was not extracted accurately when the point cloud registration did not work properly. Semi-dynamic information was not extracted twice accurately out of 1000 frames. Even if the semi-dynamic information is temporarily unavailable, the location of semi-dynamic information can be inferred from previous information, so we do not think this will be a major problem.

The accuracy of the point cloud registration should increase as the common area between the sensor data and the static map increases. Therefore, it is presumed to improve point cloud registration accuracy Therefore, it is presumed to improve point cloud registration accuracy by obtaining wider range of information using 360° LiDAR sensor. It also improves the accuracy of semi-dynamic information extraction.

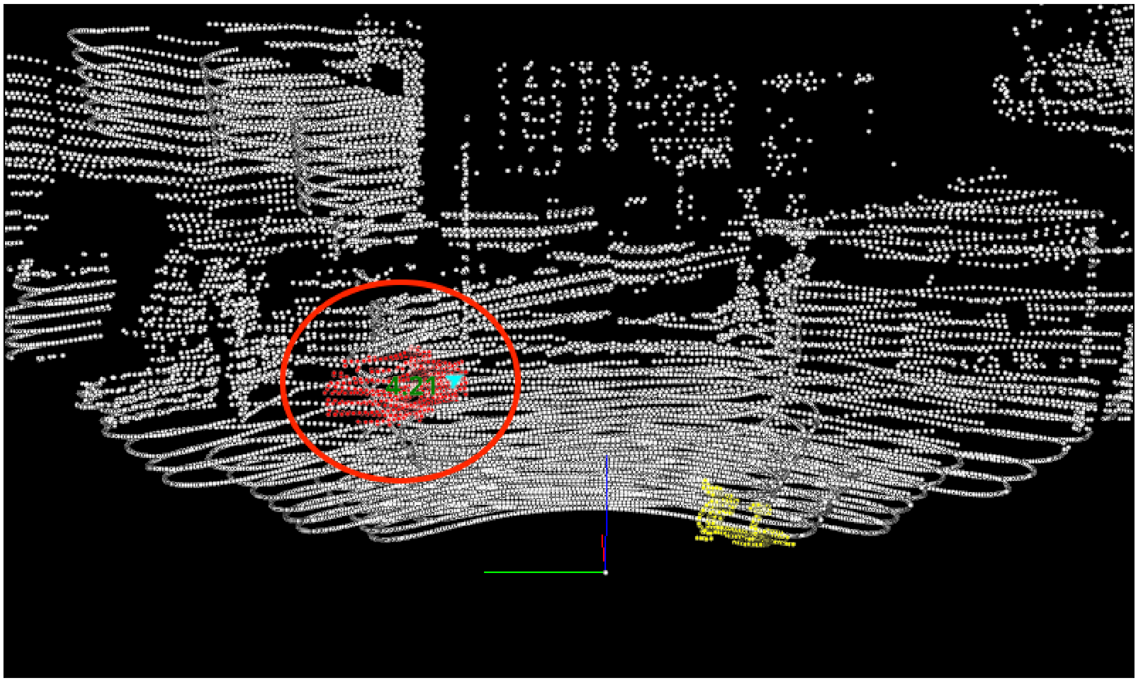
4.3.4 Velocity Vectors of Moving Objects

For dynamic information, the direction of travel is indicated by arrows and speed (meters per second) is indicated by letters. The velocity is obtained by relating the center of gravity of a moving object at time t to the nearest neighbor center of gravity of moving object at time $t-1$ as the same object. The Figure 12 shows four consecutive frames of in which a moving object is included. Figure 12b is the same as Figure 5a. The car is moving in the upper right direction in the figure. Although the speed and direction of movement are sometimes correct, they are sometimes inaccurate because they are greatly affected by alignment errors. It can be improved by analyzing in the time direction since the direction was correct in most cases.

4.4 Limitations and Further Direction

In this study, data obtained by one sensor was divided into multiple sensor data. Therefore, it was not possible to consider information from different angles of the same object, which occurs when multiple sensors are actually used. In addition, since the sensor was fixed in this study, it is necessary to verify whether the map can be created even when the sensor moves. In the future, we will conduct experiments using multiple sensors to confirm whether the map can be created even in the case of data overlapping and sensors moving.

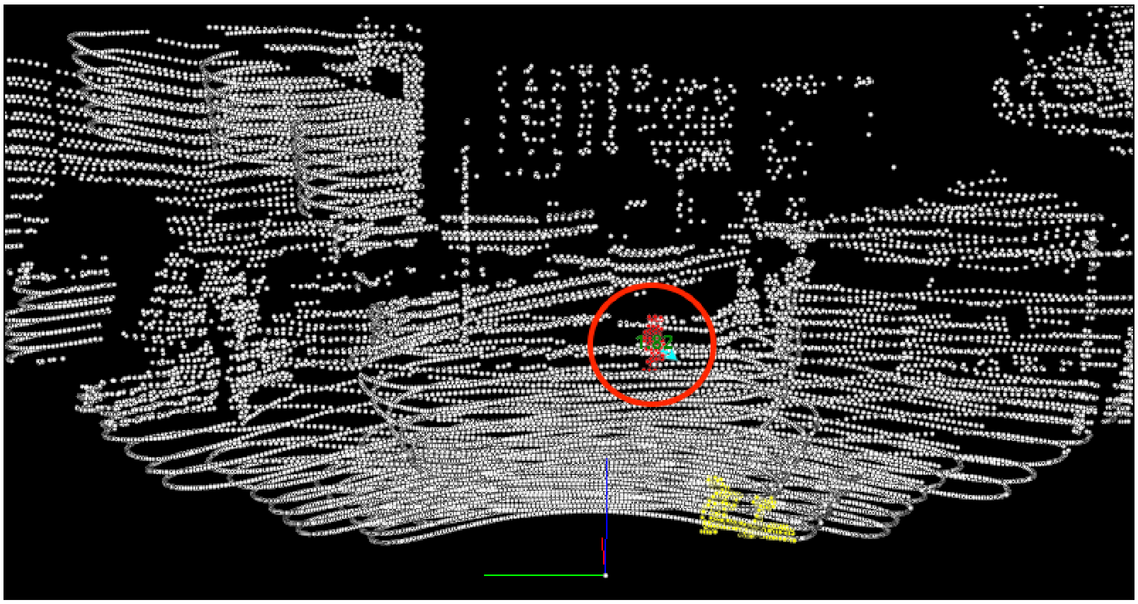
As explained in Section 2, it is assumed that the map will be created on the MEC. Therefore, it is necessary to conduct experiments including network transfer of sensor data to MEC in the future. Since dynamic information needs to be updated in real-time, we are considering the use of 5G for communication.



(a) Created point cloud map

(b) Image data at the same time

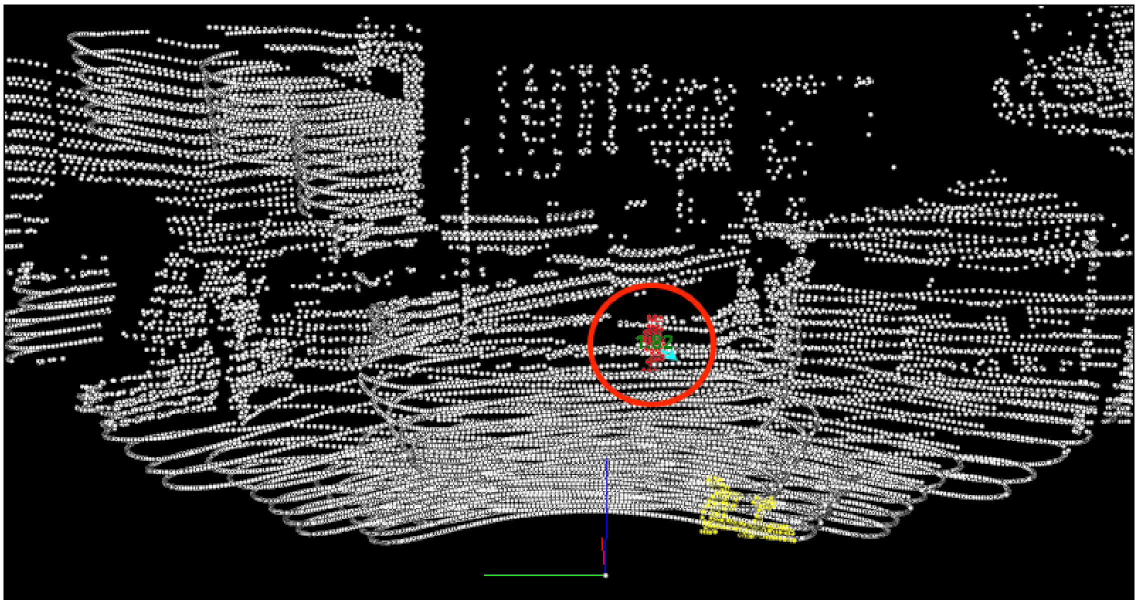
Figure 5: Visualization of a extracted moving car information



(a) Created point cloud map

(b) Image data at the same time

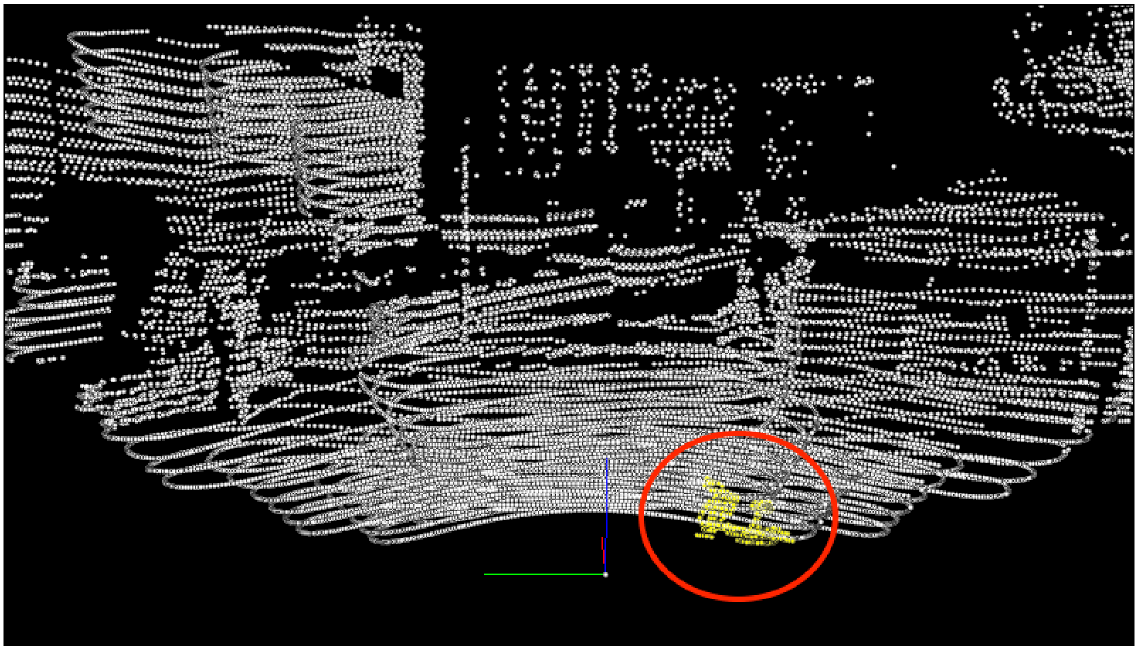
Figure 6: Visualization of a extracted moving bicycle information



(a) Created point cloud map

(b) Image data at the same time

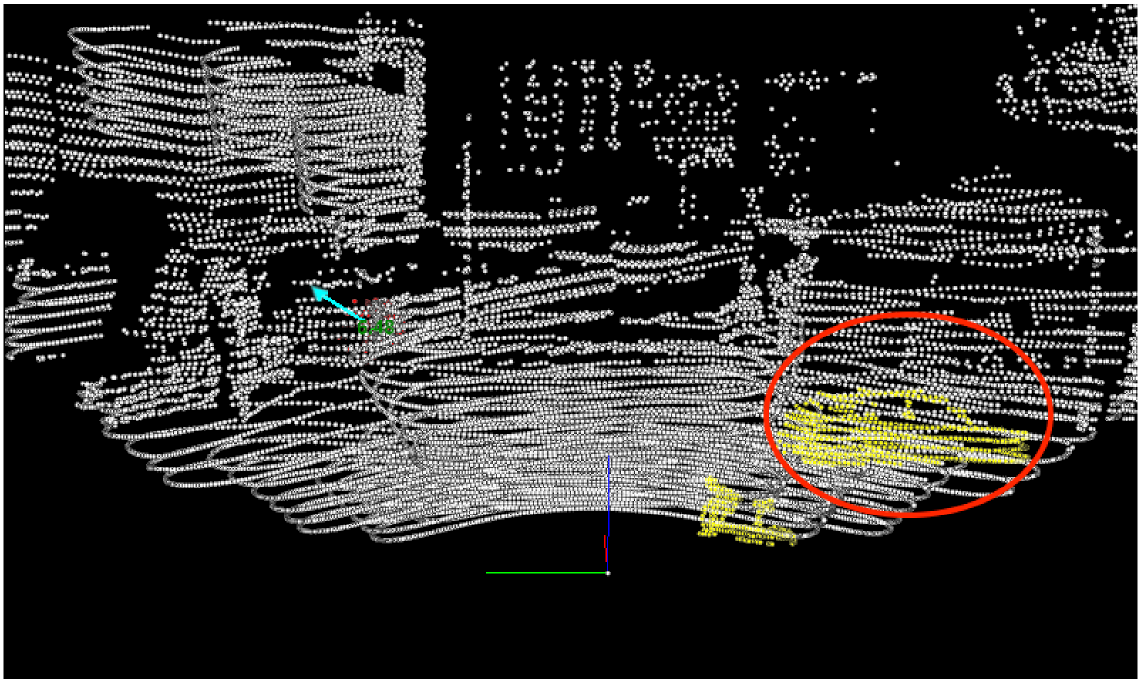
Figure 7: Visualization of a extracted pedestrian information



(a) Created point cloud map

(b) Image data at the same time

Figure 8: Visualization of a extracted parked bicycle information

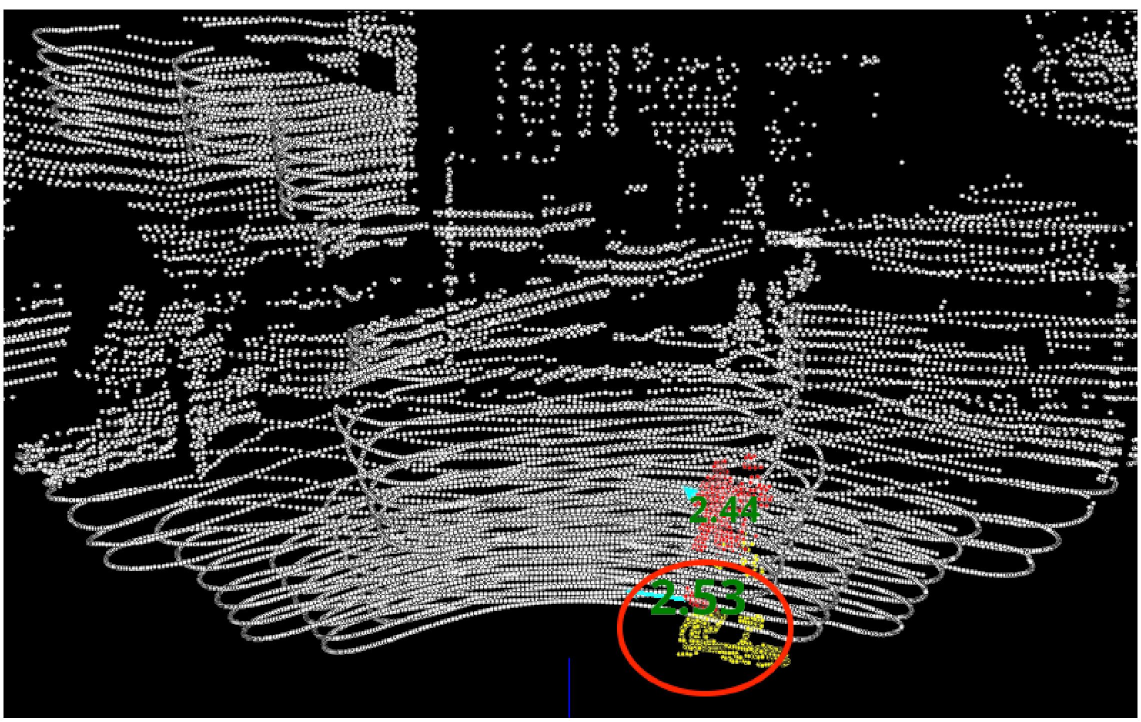


(a) Created point cloud map

(b) Image data at the same time

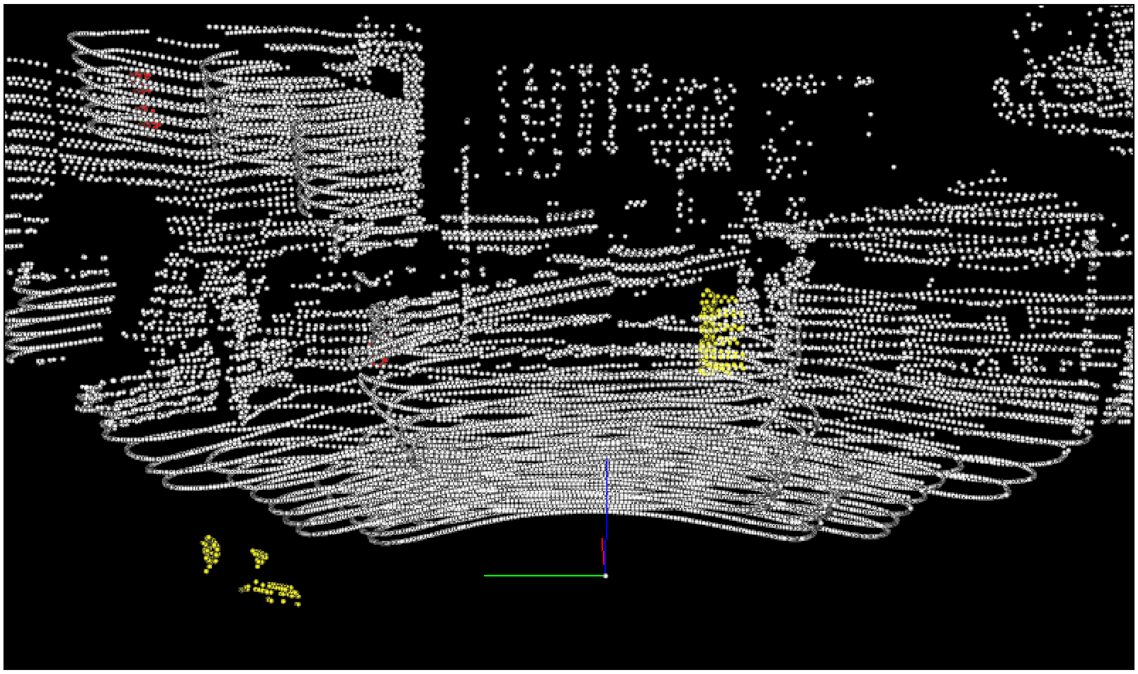
Figure 9: The case when a moving object is not detected

(a) Created point cloud map



(b) Image data at the same time

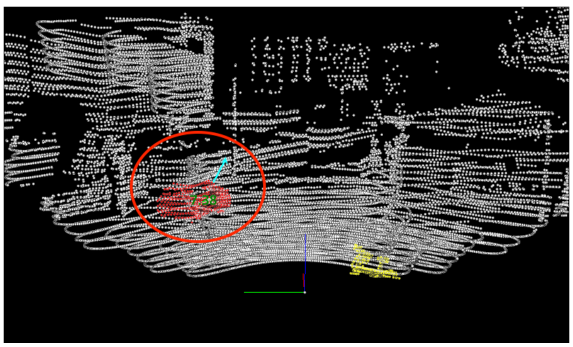
Figure 10: The case when a parked bicycle is not detected



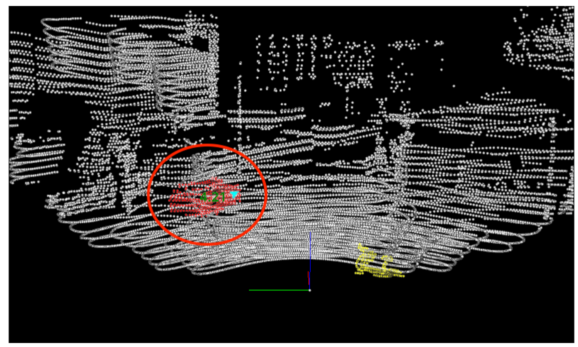
(a) Created point cloud map

(b) Image data at the same time

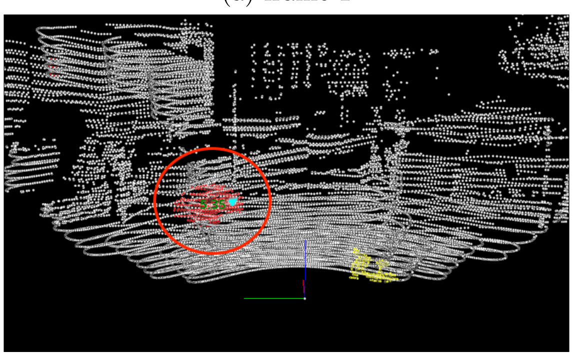
Figure 11: The case when a parked bicycle is not detected



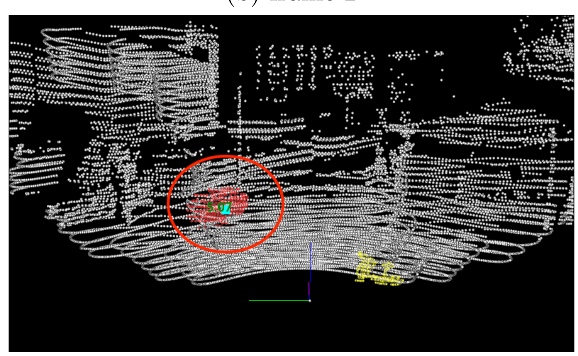
(a) frame 1



(b) frame 2



(c) frame 3



(d) frame 4

Figure 12: Velocity vectors of moving objects

5 Conclusion

In this study, we extract dynamic and semi-dynamic information from multiple sensors and integrate the extracted information on a static map. It was confirmed that dynamic information about cars, bicycles, and pedestrian can be represented on the static map. However, sometimes the detection of moving objects did not work, in which case the information was extracted as semi-dynamic information. A possible reason for the false detection of dynamic objects is that the type of LiDAR was different between the data used for LiDAR-MOS training and the data used for the experiment. The extraction of semi-dynamic information was generally successful. When the alignment was not successful, the semi-dynamic information could not be extracted, but this problem could be solved by considering the previous frames. Also, it is presumed that the alignment accuracy can be improved by using a 360° LiDAR sensor.

Future work will be required to verify whether it is possible to create maps even with overlapping data and with moving sensors using multiple LiDAR sensors. Also, it is necessary to conduct further experiments including network transmission between the MEC and the sensors. For dynamic information, it is also necessary to consider whether it can be updated in real-time.

Acknowledgments

I could not have written this paper without the help of many people. I would like to express my sincere gratitude to Associate Professor Shin'ichi Arakawa of Osaka University for his insightful advice and invaluable suggestions. Without his help, I would not have been able to overcome the difficulties in research. I also would like to express my greatest gratitude to Professor Masayuki Murata of Osaka University for his valuable comments and insights. I would like to sincerely appreciate Associate Professor Yuichi Ohsita, and Assistant Professor Daichi Kominami of Osaka University for providing many valuable comments. In addition, I also thank all the members of Murata Laboratory for their support.

References

[1] S.-W. Ko, H. Chae, K. Han, S. Lee, D.-W. Seo, and K. Huang, "V2X-Based Vehicular Positioning: Opportunities, Challenges, and Future Directions," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 144–151, 2021.

[2] H. Bagheri, M. Noor-A-Rahim, Z. Liu, H. Lee, D. Pesch, K. Moessner, and P. Xiao, "5G NR-V2X: Toward Connected and Cooperative Autonomous Driving," *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 48–54, 2021.

[3] H. Ma, S. Li, E. Zhang, Z. Lv, J. Hu, and X. Wei, "Cooperative Autonomous Driving Oriented MEC-Aided 5G-V2X: Prototype System Design, Field Tests and AI-Based Optimization Tools," *IEEE Access*, vol. 8, pp. 54 288–54 302, 2020.

[4] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5668–5677, 2021.

[5] T.-K. Lee, J.-J. Chen, Y.-C. Tseng, and C.-K. Lin, "Effect of Packet Loss and Delay on V2X Data Fusion," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 302–305.

[6] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed Dynamic Map Fusion via Federated Learning for Intelligent Networked Vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 953–959.

[7] C. Philip, D. V. Vanitha, and K. Keerthi, "Vehicle Detection and Collision Avoidance System," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2022, pp. 971–973.

[8] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, pp. 6529–6536, 2021.

[9] 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*, 2009, pp. 3212–3217.

[10] R. Rusu, Z. Marton, N. Blodow, and M. Beetz, "Persistent Point Feature Histograms for 3D Point Clouds," *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, vol. 16, 01 2008.

[11] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[12] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2724–2729 vol.3.

[13] 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*, ser. KDD'96. AAAI Press, 1996, p. 226–231.