*Proceedings of the 5th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'19)*
*Prague, Czech Republic – August, 2019*
*Paper No. MVML 107*
*DOI: 10.11159/mvml19.107*

# Research on Feature Descriptors for Vehicle Detection by LIDAR

**Takuma Oiwane, Priscilla Indira Osa, Shuichi Enokida**
Kyushu Institute of Technology
680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, Japan
oiwane_takuma753@mail.kyutech.jp; priscilla.indira.osa@gmail.com; enokida@ai.kyutech.ac.jp

***Abstract* -** LIDAR is one of remote sensing technologies to measure the distance between the sensor and objects (e.g. pedestrians, vehicles) with pulsed laser light, accurately. Because of its robustness to dynamic lighting conditions and obtainable high spatial resolution, recognition methods using LIDAR have a good capability in object recognition. This is the reason why utilizing LIDAR for autonomous vehicle and/or supporting safety driving systems. In this paper, we focus on a vehicle detection method using LIDAR, and assess the feature descriptors, including two kinds of our proposal descriptors, for point cloud data. Furthermore, we validate appropriate feature descriptors using Real AdaBoost algorithm.

***Keywords*:** Advanced driver-assistance systems (ADAS), Active Safety, LIDAR, Vehicle Detection

## 1. Introduction

In recent years, safety and convenience of automotive have been drawing increasing attention in the automotive industry. Because of that, Advanced driver-assistance systems (ADAS) and Autonomous driving system [1, 2] have been actively developed, and these systems are expected to become widespread. Camera, Millimeter-wave radar, and Laser Imaging Detection and Ranging (LIDAR) are typical sensors used to detect targets on road environments. Camera is able to acquire detailed information of the target. However, it is easily affected by environment lighting condition such as night and back light. Millimeter-wave radar is easily affected by environmental changes. However it is not good at identifying the shape and type of the targets. On the contrary, LIDAR is robust against lighting changes. In addition, it can obtain the information of distance to the target and the three-dimensional (3D) shape with high accuracy. Therefore, the utilization of LIDAR in ADAS and Automated driving is attracting attention.

There are various types of target detection using LIDAR, such as pedestrian detection [3, 4], utility pole detection [5], curb detection [6], and vehicle detection [7, 8]. According to the occurrence of traffic accidents in 2017 [9] announced by the Tokyo Metropolitan Police Department of Transport Bureau, the number of collisions between vehicles accounts for the majority of traffic accidents. Therefore, vehicle detection is an essential technology to reduce collisions between vehicles, and it is necessary to grasp the distance of nearby vehicles with high accuracy. In this paper, we focus on a vehicle detection method using LIDAR [7]. We designed the feature descriptors of point cloud data acquired by LIDAR and performed classification with Support Vector Machine (SVM). In addition, we examined appropriate feature descriptors with Real AdaBoost.

## 2. Vehicle detection by LIDAR

This chapter describes detecting method of vehicle using LIDAR. The flow of vehicle detection is as follows. First, vehicle candidates are selected from 3D point cloud data acquired by LIDAR. Next, selected vehicle candidates are calculated features using feature descriptors. Finally, the calculated features are input to the classifier. If the probability that a vehicle candidate is estimated to be a vehicle is equal to or more than a threshold, vehicle candidate regarded as a "vehicle". Each procedure is detailed in procedure by the following sections.

### 2.1. Selection of vehicle candidate data

The flow of selecting a vehicle candidate is as follows. First, the road surface is determined to detect 3D objects. Next, 3D objects are detected based on an occupancy grid map. Then, a cluster is created for point cloud data detected as 3D object. Finally, vehicle candidates are selected based on the size of the cluster.

### 2.1.1. Determination of road surface

In this paper, the road surface is detected based on RANdom SAmpling Consensus (RANSAC) algorithm [10]. The plane equation is obtained using Eq. (1):

$$ax + by + cz = d. \tag{1}$$

Since the height of the point at which LIDAR is positioned on the vehicle is known, the height from the road surface z can be calculated. Therefore, if the $z$ value satisfies Eq. (2), the estimated plane is regarded as the road surface:

$$p_{\min} \leq z \leq p_{\max} \,. \tag{2}$$

### 2.1.2. 3D object detection based on an occupancy grid map

3D object is detected based on the occupancy grid map [11]. First, the grid map of width $w$ is set on the $xy$ plane of the input data as shown in Fig. 1. Next, the variance value $\sigma_z$ of the $z$-axis of the grid is calculated for each grid. If $\sigma_z$ is equal to or more than the threshold $\tau$, 3D point cloud data in the grid is detected to be a 3D object.

### 2.1.3. Creating a cluster using Euclidean clustering

After detecting the 3D object, Euclidean Clustering [12] is performed on it. In Euclidean Clustering, if some points are within the range of a certain distance from a point, they are regarded as the same object and grouped together. Therefore, multiple clusters are created from one frame of 3D point cloud data.

### 2.1.4. Determination of vehicle candidate data based on cluster size

Vehicle candidates are selected based on the size of the created the cluster. Eq. (3) to (5) show conditions for selecting vehicle candidates:

$$h_{\min} \leq h \leq h_{\max}, \tag{3}$$
$$w_{\min} \leq w \leq w_{\max}, \tag{4}$$
$$l_{\min} \leq l \leq l_{\max}. \tag{5}$$

Here, $h$ is the height of the cluster, $w$ is the width and $l$ is the length. If all the expressions are satisfied, the cluster is selected as a vehicle candidate. Fig. 2 shows an example of vehicle candidate data selected based on cluster size.

## 2.2. Feature extraction of vehicle candidate

By using LIDAR, we can obtain the distance to detect objects and the 3D shape with high accuracy. In addition, it is possible to measure the intensity information of reflected light (hereinafter referred as "reflection intensity") of the 3D point cloud data and to estimate information such as the surface normal and object's material. In this paper, we focus on these characteristics and design feature descriptors for vehicle detection using information acquired by LIDAR. Table 1 shows the feature descriptors used in this paper: The features $f_1$ is proposed by Kidono et al. [3]. The features $f_2$ is proposed by Enokida et al. [4]. The features $f_3$ and features $f_4$ are newly proposed by the author. In this paper, features are calculated from the vehicle candidate, and the feature vector $\boldsymbol{f} = (f_1^T \, f_2^T \, f_3^T \, f_4^T)^T$is constructed. Each procedure is detailed in the following sections.

[ $f_1$ : Distribution of the reflection intensity [3] ]

By using LIDAR, it is possible to measure the reflection intensity of the 3D point cloud data. The value of reflection intensity can be varied depending on the distance to the objects and the object's material. The following three values are computed from the vehicle candidates, and the feature vector $f_1$ is constructed.

i) Mean of the reflection intensity
ii) Standard deviation of the reflection intensity

iii) Normalized histogram: the number of bin is 25 and the range of the reflection intensity is divided at equal intervals.

Table 1: Feature Descriptor for Vehicle Classification

| No. | Feature Descriptor | Dimensionality | ID |
|---|---|---|---|
| $f_1$ | Distribution of the reflection intensity | 27 | #1 ~ 27 |
| $f_2$ | Distribution of the type of the 3D shape | 40 | #28 ~ 67 |
| $f_3$ | Distribution of co-occurrence of the 3D shapes | 160 | #68 ~ 227 |
| $f_4$ | Number of points included in the local field | 10 | #228 ~ 237 |

[ $f_2$ : Distribution of the type of the 3D shape [4] ]

The rough shape of the object can be obtained from 3D point cloud data acquired from LIDAR. We explain how to calculate features as follows. First, vehicle candidates are divided into $N_e$ at equal intervals in the height direction from the road surface. By dividing the area, we obtain the local shape of the vehicle candidate (hereafter, the divided local area is called "*slice*"). Next, we use a particular point and its surrounding point clouds to compute eigenvalues in the each *slice*. To compute eigenvalues, we compute a variance-covariance matrix from the set of point clouds within a region of radius $r_e$ surrounding the particular point. Then, we apply Principal Component Analysis (PCA) to the variance-covariance matrix and obtain eigenvalues $\lambda_1, \lambda_2, \lambda_3$ $(\lambda_1 \geq \lambda_2 \geq \lambda_3)$. After that, we compute three values $S_0^i, S_1^i, S_2^i$ as in Eq. (6) to (8) (where $\alpha$ and $\beta$ are correction coefficients):

$$S_0^i = \lambda_1 - \alpha\lambda_2, \tag{6}$$

$$S_1^i = \lambda_2 - \lambda_3, \tag{7}$$

$$S_2^i = \beta\lambda_3. \tag{8}$$

If the value $S_0^i$ is the maximum among the three values, we judge the shape of the particular point to be “pole” shaped; if $S_1^i$ is the largest, we consider the point to be “plane” shapes; if $S_2^i$ is the largest, we consider the point to be “solid figure” shape. If the point cloud $i^*$ surrounding the particular point satisfies $(|i^* \;| \leq 1)$, the point is determined to be “unclassified”. Fig. 3 shows the distribution of the appearance frequency of the 3D shape in each slice. Therefore, we utilize the distribution as the feature and construct the feature vector $f_2$.

[ $f_3$ : Distribution of co-occurrence of the 3D shapes]

In this feature descriptor, multiple local shapes are acquired, and their co-occurrence relationship is calculated. We explain how to calculate features as follows. First, vehicle candidates are divided into $N_c$ at equal intervals in the height direction from the road surface. Next, we use the particular point in the each *slice* and its surrounding point clouds to compute eigenvalues. To compute eigenvalues, we consider the set of point clouds within a region of two radius $r_e$ and $r_c$ surrounding remarkable point, and we compute a variance-covariance matrix. Then, we apply PCA to the variance-covariance matrix and obtain eigenvalues $\lambda_1, \lambda_2, \lambda_3$ $(\lambda_1 \geq \lambda_2 \geq \lambda_3)$. After that, we compute three values $S_0^i, S_1^i, S_2^i$ as in Eq. (2.6) to (2.8). The shape estimation method is the same as $f_2$. Fig. 4 shows the distribution of co-occurrence of the 3D shapes in each slice. Therefore, we utilize the distribution as the feature and construct the feature vector $f_3$.

[ $f_4$ : Number of points included in the local field ]

The uniqueness of the vehicle’s 3D point cloud distribution is the body of the vehicle reflects the light from the sensor while the glass is passed through. Therefore, we considered that a characteristic distribution of point cloud could be obtained from the roof to the tires of the vehicle. In this feature descriptor, we calculate data for each slice as the feature. We explain how to calculate the feature as follows. First, the total score of point cloud data is obtained from the vehicle candidate. Next, the vehicle candidate is divided into $N_c$ at equal intervals in the height direction from the road surface. Finally, the score of point cloud data is calculated from each slice, and the value is normalized with the total score of the vehicle candidate calculated in the previous step. Fig. 5 shows the score of point cloud data is obtained from each slice. We utilize the distribution as the feature and construct the feature vector $f_4$.

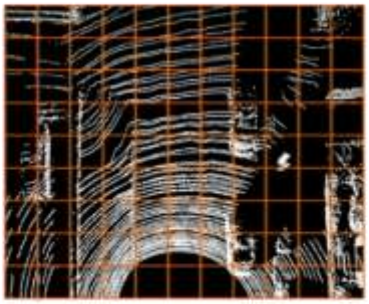
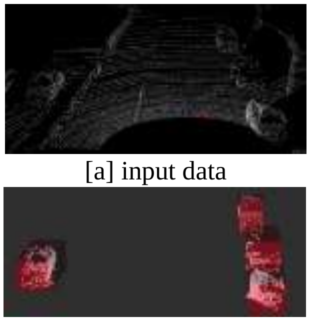


Fig. 1: Example of point cloud data projected into $xy$ plane.



[a] input data

[b] output data

Fig. 2: Example of vehicle candidate data based on cluster size.

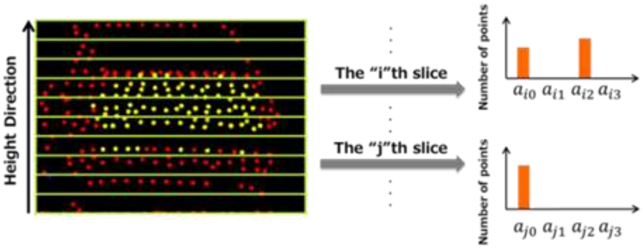


Fig. 3: Distribution of the type of the 3D shape.

### 2.3. Vehicle detection by classifier

Vehicle detection is performed by inputting the calculated the feature vectors to the trained classifier. Vehicle data as the positive data and non-vehicle data as the negative data are used to train the classifier. Fig. 5 shows examples of training data corresponding to vehicle data and non-vehicle data. Vehicle data includes vehicles taken from various angles, and non-vehicle data includes signs, plants, parts of buildings, etc. In this paper, we utilized SVM [13] and Real AdaBoost [14] as classifiers. The following describes each classifier in detail.

#### 2.3.1. SVM-based classifier

SVM is supervised learning model used for classification and regression analysis. The classification boundary is constructed to maximize the margin, which is the distance between the support vector in the training data and the classification boundary. Once the classification boundary is constructed, binary classification is performed. The advantages of SVM are the following: the accuracy is good even when there is little learning data or the feature dimension is large; generalization capability to classify new data is high. On the other hand, the disadvantages of SVM are the following: the cost of calculation increases as the amount of learning data increases; since it is a black-box model in which the input-output relationship is not explicit, it is difficult to interpret which features affect the identification result and performance.

#### 2.3.2. Real AdaBoost-based classifier

Real AdaBoost is an ensemble learner that constructs a strong classifier with high classification ability by combining weak classifiers with low classification ability sequentially. The advantages of Real AdaBoost are the following: since the output of weak classifiers is a real number, it is possible to respond flexibly to changes in the input, leading to improved accuracy; since the degree of classification with respect to the test data can be grasped, it is possible to interpret which features lead to the classification result and analyze the identification performance. On the other hand, the disadvantages of Real AdaBoost is that it is necessary to determine the number of learning experimentally.

In the learning process, among the weak classifier candidates found in each learning, the one with the highest evaluation value is selected as the weak classifier. Here, one weak classifier indicates one feature vector. In this paper, we focus on the weak classifiers and consider them as good features for classification.

## 3. Experiments

To confirm the effectiveness of the feature descriptors for vehicle detection, the quantitative evaluation on 3D point cloud data was carried out. First, the identification performance by SVM was examined. Next, we utilized Real AdaBoost to interpret which features affect to the identification results and analyzed the performance. Finally, the identification performance by SVM where only the features selected by Real AdaBoost was examined.

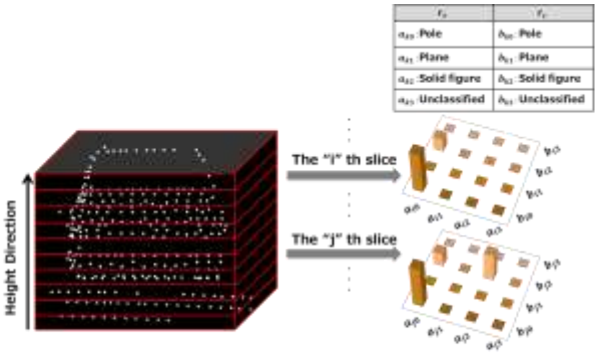


Fig. 4: Distribution of co-occurrence of the 3D shapes.

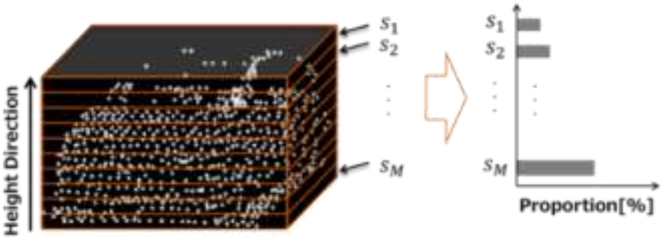


Fig. 5: Number of points included in the local field.

[a] positive data (vehicle)

[b] negative data (non-vehicle)

Fig. 6: Examples of learning and evaluation data.

Fig. 7: Data measurement using LIDAR and experimental car.

### 3.1. Experimental condition

Fig. 7 shows high resolution LIDAR sensor device named "HDL-64E-S3" manufactured by Velodyne Inc. Experiment data were obtained in a real road environment by LIDAR. Experiment data were obtained while driving around the Kyushu Institute of Technology, Iizuka Campus, and manually labelled data was used for learning and evaluation of the classifier. In order to evaluate the identification performance according to the distance, the data of the vicinity of 10m (5-15m) and the vicinity of 30m (25-35m) from LIDAR are prepared respectively. Table 2 shows the number of experimental data for each distance. The Point Cloud Library (PCL) [16] was used as a library to handle 3D point cloud data, and the parameters in Table 3 were used in the experiment.

### 3.2. Experiment of identification performance by SVM

In this section, the identification performance by SVM was examined. We used SVM with the Radial Basis Function (RBF) kernel as the classifier, and Receiver Operating Characteristic (ROC) curve obtained by 10-fold cross validation was used for evaluation.

First, we checked the identification performance when using feature descriptor one by one. Fig. 8 shows the experimental results. From the results, it was confirmed that $f_4$ shows the highest performance at all distances. In addition, it was confirmed that the identification performance of all feature descriptors become worse as the detection target is in the farther distance. Next, we examined the identification performance when using feature descriptors in combination. Fig. 9 shows the method used for comparison and Fig. 10 shows the experimental results. From the results, it was confirmed that the methods 8, 9 and 11 using $f_4$ show high identification performance at all the distances.

Based on the experimental results, we can conclude that the identification performance was improved by combining $f_4$ (which showed the highest performance in the first experiment) with other features. In addition, the combination of feature descriptions using different features such as $f_1$ and $f_3$ as in method 2 led to improved performance. On the other hand, the combination of feature descriptors using the same type shape features of $f_2$ and $f_3$ as in Method 4 did not lead to performance improvement. Therefore, improvement of identification performance can be expected by combining different type of features together.

Table 2:  Number of experimental data.

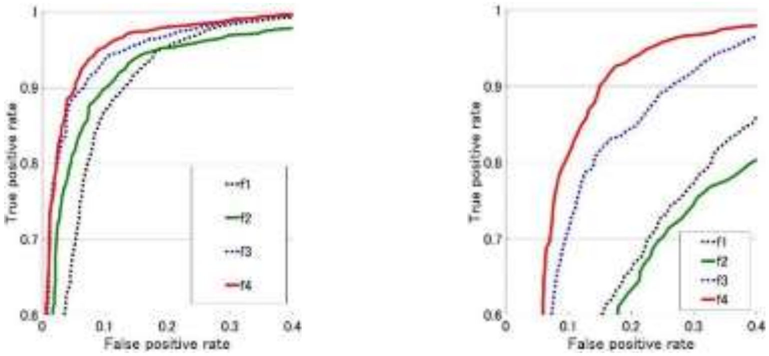
| data | 10m | 30m |
|---|---|---|
| Vehicle data | 1,925 | 1,326 |
| Non-vehicle data | 1,216 | 1,025 |

Table 4: Evaluation of identification performance each discriminator.

| F-value | 10m | 30m |
|---|---|---|
| SVM | 0.941 | 0.883 |
| Real AdaBoost | 0.973 | 0.894 |

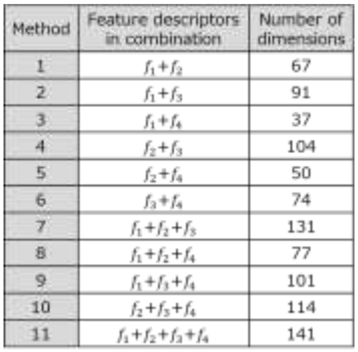
Table 3: Parameters used in experiment.

| Parameter name | Value | Section | Parameter name | Value | Section | Parameter name | Value | Section |
|---|---|---|---|---|---|---|---|---|
| $p_{\min}$ | 1.7 | 2.1.1. | $h_{\max}$ | 2.5 | 2.1.3. | $\alpha$ | 5.0 | 2.1.3 |
| $p_{\max}$ | 1.9 |  | $w_{\min}$ | 1.0 |  | $\beta$ | 10.0 |  |
| σ | 0.35 | 2.1.2. | $w_{\max}$ | 2.5 |  | $N_c, M$ | 10 |  |
| $t_z$ | 0.05 |  | $l_{\min}$ | 0.0 |  | $r_e$ | 0.35 |  |
| $h_{\min}$ | 1.0 | 2.1.3. | $l_{\max}$ | 5.0 |  | $r_c$ | 2.5 $r_e$ |  |



[a] 10m [b] 30m

Fig. 8: Results of performance tests for various feature quantities.



| Method | Feature descriptors in combination | Number of dimensions |
|---|---|---|
| 1 | $f_1+f_2$ | 67 |
| 2 | $f_1+f_3$ | 91 |
| 3 | $f_1+f_4$ | 37 |
| 4 | $f_2+f_3$ | 104 |
| 5 | $f_2+f_4$ | 50 |
| 6 | $f_3+f_4$ | 74 |
| 7 | $f_1+f_2+f_3$ | 131 |
| 8 | $f_1+f_2+f_4$ | 77 |
| 9 | $f_1+f_3+f_4$ | 101 |
| 10 | $f_2+f_3+f_4$ | 114 |
| 11 | $f_1+f_2+f_3+f_4$ | 141 |

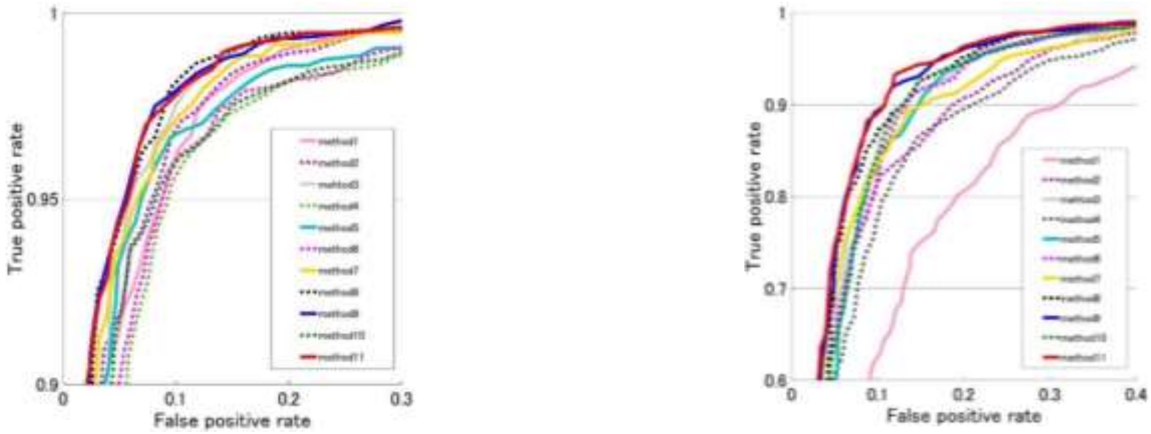
Fig 9: The method used in Experiment 3.2.

### 3.3. Experiment of identification performance by Real AdaBoost

In this section, the identification performance by Real AdaBoost was verified. In this experiment, Real AdaBoost was trained 500 times and evaluation was performed by 10-fold cross validation.

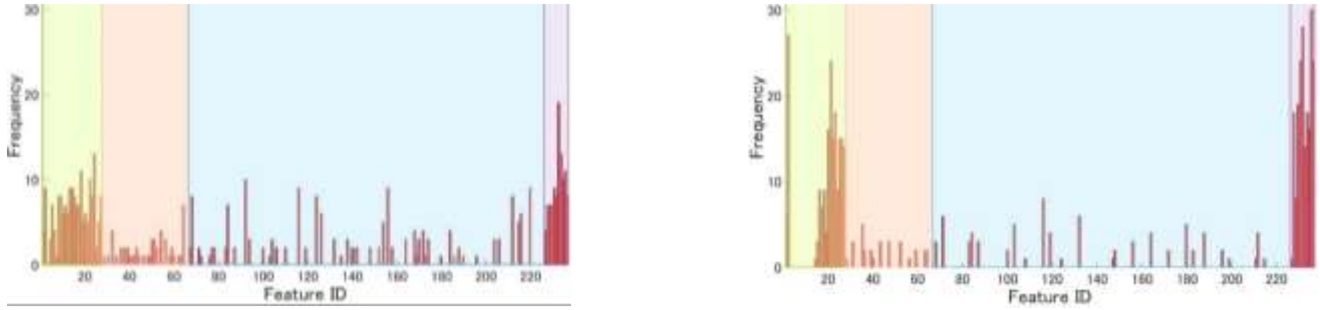
First, we compared the identification performance of the classifier based on SVM and the classifier based on Real AdaBoost. SVM used all feature descriptors shown in Table 1, and Real AdaBoost used only automatically selected features. For evaluation, we used the F-value, which is the harmonic mean of recall and precision. Table 4 shows the experimental results. From the results, it is confirmed that the classifier by Real AdaBoost shows performance equal to or better than the classifier of SVM. Therefore, it is expected to show equal or better performance while maintaining the generalization capability of SVM by using only the feature automatically selected by Real AdaBoost for identification. Next, we examined the tendency of feature selection based on Real AdaBoost in the above experiment. Fig. 11 shows the learning results for each distance. From the results, it was confirmed that the features selected at each distance are different.

Based on the experimental results, we found the selection tendency of each feature descriptor. The feature descriptor $f_1$, in the near distance, was selected in well-balanced way regardless whether the reflection intensity is low or high. On the other hand, in distant area, the part with lower reflection intensity was not selected and the part with high reflection was selected. The feature descriptors $f_2$ and $f_3$ were selected differently depending on the local area of the cluster, and the features that are not selected increased in the distant area. The feature descriptor $f_4$ was selected with well-balance in each local area regardless of the distance. From the above, by using Real AdaBoost, it became easy to analyze the features leading to the identification results, and the existence of the features selected at each distance became clear.



[a] 10m [b] 30m

Fig. 10: Results of performance tests for various method quantities.



[a] 10m [b] 30m

Fig. 11: Frequency of feature selection by Real AdaBoost.

### 3.4. Evaluation experiment of classifiers constructed based on feature selection

This section, we performed an evaluation experiment of classifiers constructed with features selected based on Real AdaBoost. For verification, we prepared three classifiers, as follows: Classifier1 utilized all feature descriptors to learn the classification rule (237 dimensions). Classifier2 utilized features selected at least once with Real AdaBoost in Section 3.3 to learn the classification rule (113 dimensions). Classufuer3 utilized feature descriptors $f_1$ and $f_4$ to learn the classification rule (37 dimensions). In the experiment, the discrimination performance of three prepared classifiers was evaluated by utilizing raw data obtained in the vicinity of 10m. The ROC curve obtained by 10-fold cross validation was used for evaluation. Fig. 12 shows the experimental results. The results show that Classifier2 has higher identification performance than Classifier3. Furthermore, Classifier2 showed almost same identification performance as Classifier1. From the above results, it is expected to construct a high-performance classifier by using only features that are effective for classification. In addition, it is expected to maintain identification performance while reducing the number of dimensions by considering feature selection based on Real AdaBoost.

## 4. Conclusion

In this paper, we focus on a vehicle detection method using LIDAR and assessed the feature descriptors for point cloud data acquired by LIDAR. Furthermore, we examined the appropriate feature descriptors using Real AdaBoost algorithm. From the experimental results, it was confirmed that improvement of identification performance can be expected by combining feature descriptions with different type of features. In addition, by using Real AdaBoost, it became easy to analyze the features leading to the good identification performance, and the existence of the features selected at each distance became clear. Furthermore, it can be expected to maintain identification performance while reducing the

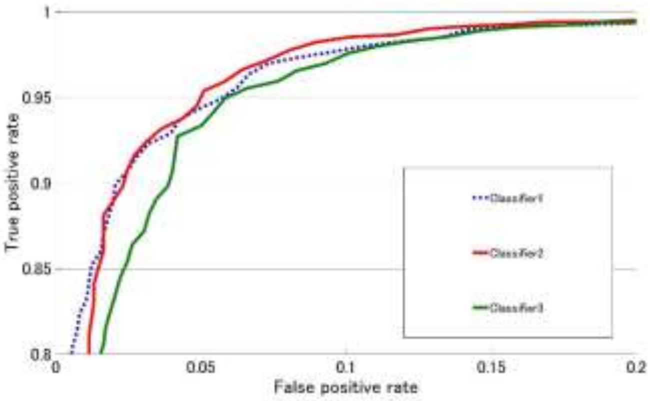


Fig. 12: Results of performance tests for at distance 10m.

number of dimensions by considering feature selection based on Real AdaBoost. From the above, we expect that it will be an indicator to design feature descriptors for vehicle detection using LIDAR.

In the future, we plan to examine the new feature descriptor proposal such as interpolation of the point cloud.

## References

[1] M. Buehler, etc., “The DARPA Urban Challenge,” *Autonomous Vehicles in City Traffic*, 2010.

[2] A. Broggi, etc., “Sensing Requirements for a 13,000 km Intercontinental Autonomous Drive,” *IEEE Intelligent Vehicles Symposium*, pp. 500-505, 2010.

[3] K. Kidono, etc., “Pedestrian Recognition Using High-definition LIDAR,” *IEEE Intelligent Vehicles Symposium*, no. 4, pp. 405-410, 2011.

[4] S. Enokida, etc., “Pedestrian detection based on the shape feature and the reflection intensity distribution using LIDAR,” *Fuzzy System Symposium*, pp. 141-146, 2016.

[5] H.Yokohama, etc., “Detection and Classification of pole-like Objects from Mobile Laser Scanning Data of Urban Environments,” *International Journal of CAD/CAM*, vol. 13, no. 2, pp. 31-40, 2013.

[6] M. Jono, etc., “Curb detection from 3D point cloud acquired by LiDAR based on Weighted Markov Chain Model,” *International Conference on Information and Communication Technology Robotics*, 2016.

[7] M. Himmelsbach, etc., “LIDAR-based 3D Object Perception,” *IEEE International Conference on Robotics and Automation*, pp. 1-7, 2008.

[8] Bo Li, etc., “Vehicle Detection from 3D Lidar Using Fully Convolutional Network,” *Robotics: Science and Systems*, 2016.

[9] “Occurrence of traffic accidents in 2017,” Tokyo Metropolitan Police Department-traffic Bureau [Online]. Available: https://www.npa.go.jp/publications/statistics/koutsuu/H29zennjiko.pdf

[10] M. A. Fischler etc., “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[11] S. Thrun, etc., “Probabilistic Robotics,” *The MIT Press*, 2005.

[12] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,” *KI - Künstliche Intelligenz*, vol. 24, no. 4, pp. 345-348, 2010.

[13] N. Cristianini, etc., “An Introduction to Support Vector Machines,” *Cambridge University Press*, 2000.

[14] Robert E. Schapire etc., “Improved Boosting Algorithms Using Confidence-rated Predictions,” *Machine Learning*, vol. 37, issue 3, pp. 297-336, 1999.

[15] Y. Yamauchi, etc., “People Detection Based on Co-occurrence of Appearance and Spatiotemporal Features,” *Progress in Informatics*, no. 7, pp.33–42, 2010.

[16] “PCL ‐ Point Cloud Library,” [Online]. Available: http://pointclouds.org