

## CLASSIFICATION OF POLE-LIKE OBJECTS USING POINT CLOUDS AND IMAGES CAPTURED BY MOBILE MAPPING SYSTEMS

Yuma Mori, Keisuke Kohira, Hiroshi Masuda

The University of Electro-Communication, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan –  
(yuma.mori, keisuke.kohira, h.masuda)@uec.ac.jp

Commission II, WG II/4

**KEY WORDS:** Mobile Mapping System, Point-Cloud, Pole-Like Object, Machine Learning, Convolutional Neural Network

### ABSTRACT:

The vehicle-based mobile mapping system (MMS) is effective for capturing 3D shapes and images of roadside objects. The laser scanner and cameras on the MMS capture point-clouds and sequential digital images synchronously during driving. In this paper, we propose a method for detecting and classifying pole-like objects using both point-clouds and images captured using the MMS. In our method, pole-like objects are detected from point-clouds, and then target objects, which are objects attached to poles, are extracted for identifying the types of pole-like objects. For associating each target object with images, the points of the target object are projected onto images, and the image of the target object is cropped. Each pole-like object is represented as a feature vector, which are calculated from point-clouds and images. The feature values of a point-cloud are calculated by point processing, and the ones of the cropped image are calculated using a convolutional neural network. The feature values of point-clouds and images are unified, and they are used as the input to machine learning. In experiments, we classified pole-like objects using three methods. The first method used only point-clouds, the second used only images, and the third used both point-clouds and images. The experimental results showed that the third method could most accurately classify pole-like objects.

### 1. INTRODUCTION

The vehicle-based mobile mapping system (MMS) is effective for capturing shapes and images of road environment. The MMS is a vehicle, on which laser scanners, cameras, GPSs and IMU are mounted, as shown in Figure 1. The MMS can capture dense point-clouds and sequential digital images synchronously during driving. Since point-clouds and images include various types of objects, such as utility poles, traffic signals, and trees, it is often required to segment and classify each object. Especially, detection and classification of pole-like objects are practically important, because many pole-like objects are placed on roadsides, and they have to be repeatedly maintained.

Golovinskiy, et al., Weinmann et al., Zhu et al., used point-clouds to extract road-side objects and classified them using supervised machine learning. However, their research is a rough classification of feature classes, and the classified types of pole-like objects were limited (Golovinsky et al., 2009; Zhu et al., 2010; Weinmann et al, 2014). Fukano, et al. proposed a method for segmenting and classifying pole-like objects from point-clouds (Fukano et al., 2015). Their method could achieve good recognition rates, but it sometimes misclassified objects. Some failure cases are shown in Figure 2. When the vehicle runs at a high speed, scan-lines on objects become sparse, as shown in Figure 2(a). In such cases, it is difficult to distinguish the difference of object shapes using point-clouds. In addition, if the shapes of objects are similar as shown in Figure 2(b), objects may be incorrectly classified.

In recent years, image-based object recognition based on a convolutional neural network (CNN) has been greatly improved. Zhu et al. classified traffic signs into ten or more classes using CNN (Zhu, et al., 2016a, Zhu, et al., 2016b). However, their method focused on only traffic signs.

The image-based classification is effective only if objects are correctly segmented from images. In our evaluation, segmentation failed in some cases if the object color was similar to the background color or if the image was backlighted, as shown in Figure 3(a). The classification also failed when object colors were faded, as shown in Figure 3 (b).

Point-clouds maintain 3D shapes, while images maintain object colors. Therefore, it is reasonable to use both point-clouds and digital images to correctly classify objects. Soilán, et al. detected traffic signs from point-clouds and images (Soilán, et al., 2016). Since point-clouds of traffic signs have very high intensity values, traffic signs can be easily and robustly extracted from point-



(a) Mobile mapping system



(b) Equipment on the MMS

Figure 1. Mobile mapping system

\* Corresponding author

clouds. By projecting points on images and cropping traffic signs on the images, they could achieve very high recognition rate. However, their method was only be applied to traffic signs, and classified using only images. Their method is similar to our method, but our method classifies objects using both point-clouds and images.

In this paper, we propose a new classification method for pole-like objects by using both point-clouds and images, which are synchronously captured by an MMS. In our method, feature values are calculated from both point-clouds and images. Feature values are combined and they are used for machine learning.

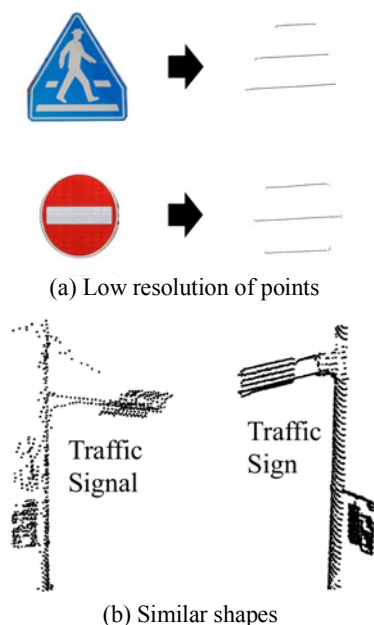


Figure 2. Failure cases of point-based object recognition.

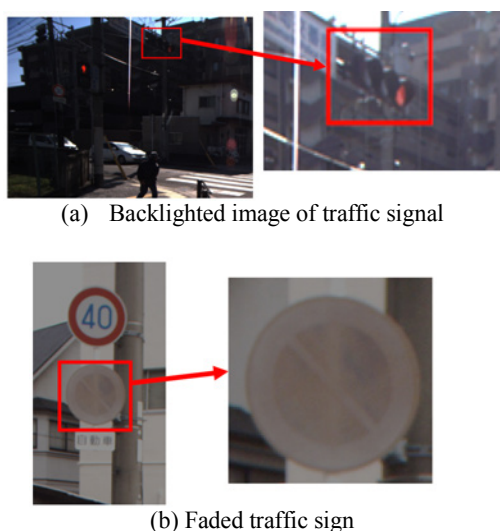


Figure 3. Failure cases of image-based object recognition.

## 2. OVERVIEW OF OBJECT CLASSIFICATION

Since pole-like objects commonly have cylindrical poles, objects attached to the poles are important for differentiating pole-like objects. The object attached to a pole is called *target objects* in

this paper. Target objects include signals, signs, lights, cables, and so on.

The MMS used in this research is the Mitsubishi MMS-X (Mitsubishi Electric, 2018), as shown in Fig. 1. The laser scanner

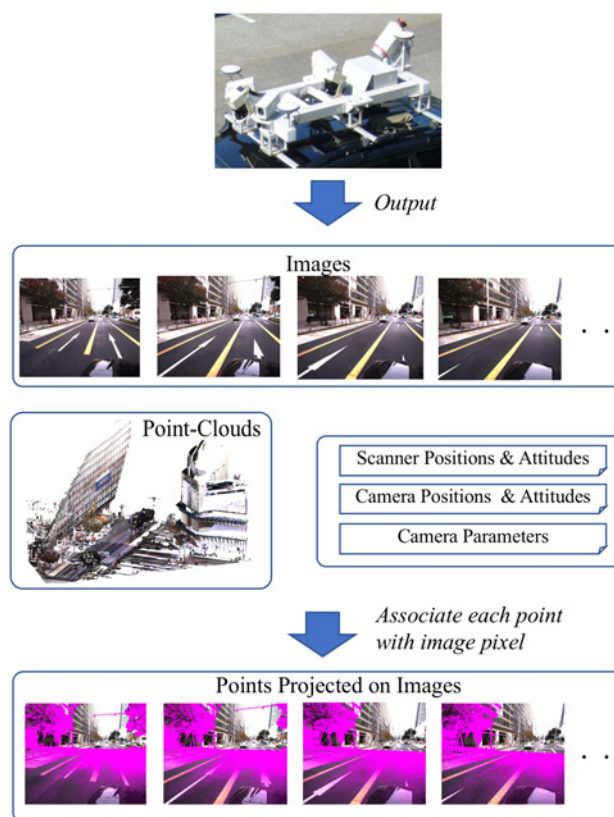


Figure 4. Output of the MMS

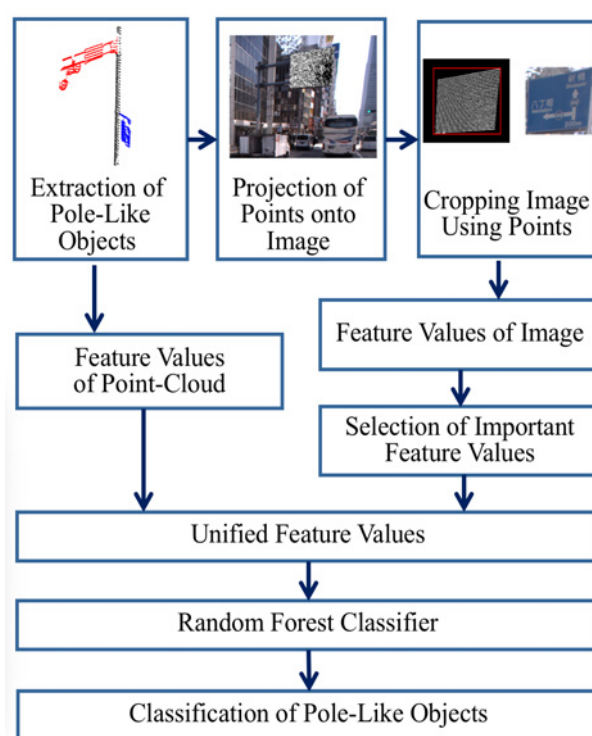


Figure 5. Process of object classification

on the MMS is RIEGL VQ 250. The rotational frequency and the scan rate of this laser scanner is 100Hz and 300,000 measurements, respectively. Point-clouds and digital images are captured during driving. Each point and each image have the GPS time, which is the time sent from satellites and represents when the point was captured.

Figure 4 shows the output from the MMS. The digital cameras output sequential images captured at certain time intervals. The laser scanner outputs point-clouds, in which 3D coordinates, intensity values, and GPS times are included. The MMS also outputs positions and attitudes of the cameras and the laser scanners during driving. The camera parameters for each camera are given in the setting files. By using these data, it is possible to associate each point in a point-cloud with a pixel of an image, as shown in Figure 4.

The process of our classification method is shown in Figure 5. The input data are point-clouds and digital images. First, cylindrical poles are extracted from point clouds. When poles are detected, target objects are extracted from points near the poles. Then feature values are calculated using poles and target objects. Feature values consist of sizes, relative positions, eigenvalues of the principal component analysis (PCA), and so on. Feature values are calculated for each pole-like object.

Feature values are also calculated from digital images. Since points and images are synchronously captured and the relative position between the laser scanner and the digital camera is given, points of each target object can be projected on an image using the pinhole camera model. Then, the region of each target object is identified and it is cropped from the image. The feature values of the cropped image are automatically calculated using CNN.

In our method, point-based feature values and image-based feature values are combined, and the unified feature values are used for machine learning. Since the number of image-based feature values is much larger than the number of point-based feature values, the dimension of image-based feature values is reduced using the importance of features. We apply the Random Forest classifier (Breiman, 2001) to image-based feature values, and calculate the importance of each feature. Only features with high importance are selected, and they are combined with point-based feature values. Finally, we classify pole-like objects using the unified feature values.

### 3. DETECTION OF POLES AND TARGET OBJECTS

#### 3.1 Detection of Poles

Fukano, et al. proposed a method for stably extracting pole-like objects from point-clouds captured using the MMS (Fukano, et al., 2015). In this paper, we detect pole-like objects using this method.

First, a point-cloud is converted to a wireframe model, in which points are connected by edges. When the rotational frequency of the laser scanner is denoted as  $f$ , neighbour points between scanlines are captured every  $1/f$  second, as shown in Figure 6 (Masuda, et al., 2015; Kohira, et al. 2017). The nearest points on the next scan-line are detected using the GPS time of each point, and wireframe models are generated by connecting neighbour points.

Figure 7 shows a process for detecting a pole. When a wireframe model is generated from points, it is sliced using horizontal

planes. In our method, the interval between horizontal planes is set to 10 cm. Intersection points are calculated between the wireframe and each horizontal plane. Then a circle is fitted to the intersection points, because the section shape of a pole is a circle. Poles are extracted using circles that are vertically aligned.

The shape of a pole is typically a cylinder or a cone. In our method, the taper of a pole can be calculated using the radii of circles. The taper is calculated as the change rate of the radius of a pole in the height direction. The taper is used as one of shape features for recognition.

#### 3.2 Detection of Target Objects

When poles are extracted, they are eliminated from the wireframe model. Then the wireframe model is subdivided into connected components, and target objects are extracted as connected components near the pole. In our method, a connected component is selected as a target object only if it is located at a height of 1.5 m or more from the bottom of the pole. Figure 8 shows examples of target objects.

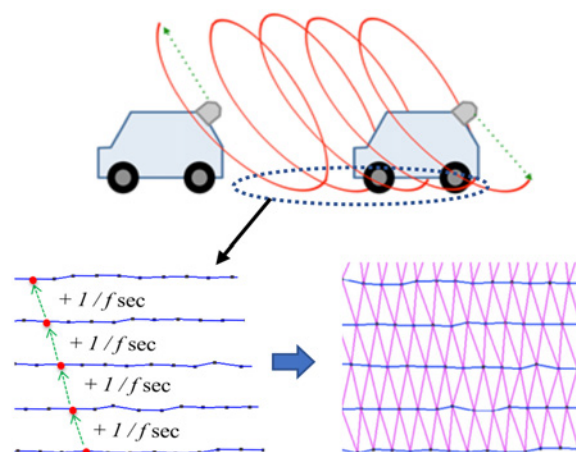


Figure 6. Conversion to wireframe model

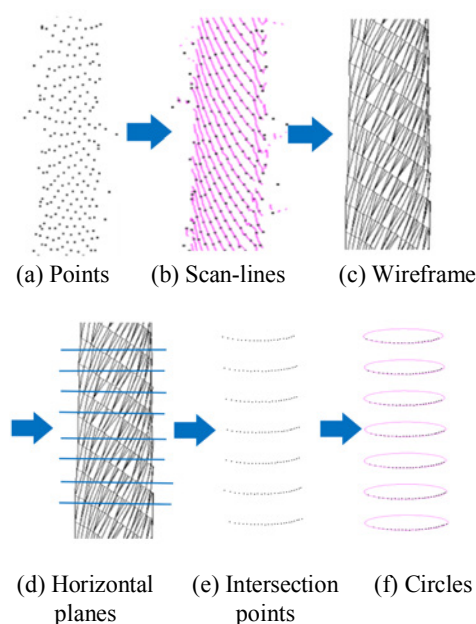


Figure 7. Detection of pole



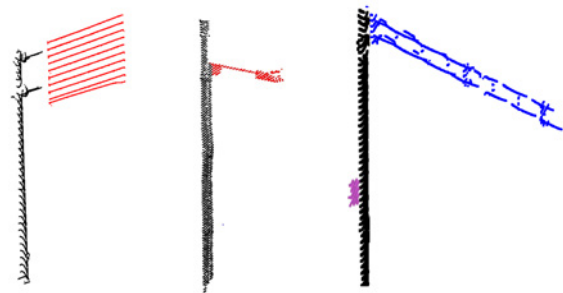


Figure 8. Target objects near poles

#### 4. FEATURE VALUES OF 3D POINT-CLOUD

In order to apply machine learning for classification, we represent each pole-like object as a set of feature values. In our method, feature values are calculated using point-clouds of target objects and poles.

Table 1 shows feature values of the point-cloud of a pole-like object. From a target object, a bounding box and eigenvalues of PCA are calculated. Figure 9(a) shows the bounding box of a target object. The bounding box is defined so that the bottom face is horizontal and the width  $a$  is the size in the maximum principal component direction for xy coordinates.

The eigenvalues of the target object are obtained by applying PCA to the 3D coordinates of the points. We denote the eigenvalues as  $\lambda_1, \lambda_2, \lambda_3$  ( $\lambda_1 \geq \lambda_2 \geq \lambda_3$ ), and their ratios as  $r_1 = \lambda_1/\lambda_2, r_2 = \lambda_2/\lambda_3$ , and  $r_3 = \lambda_3/\lambda_1$ . Also let  $\theta$  be the angle of the maximum principal component direction from the horizontal plane.

In addition, local shapes are calculated using PCA. For each point, neighbour points are detected within a certain distance. In our research, the distance was specified as 30 cm. Eigenvalues and their ratios are calculated using the neighbour points. We introduce thresholds  $\alpha$  and  $\beta$ , and define a point to be linear when  $r_1 \geq \alpha$ , planar when  $r_1 < \alpha$  and  $r_2 \geq \beta$ , and spherical otherwise. In Table 1,  $r_l, r_p$ , and  $r_s$  are ratios of numbers of linear, planar, and spherical points. In this research, we specified  $\alpha = 6$  and  $\beta = 5$ . These values were determined experimentally.

Since each point has an intensity value, the average of intensity values is also added to feature values of the target object. Figure 9(b) shows feature values for a pole. The diameters of the bottom and top circles are denoted as  $D_1$  and  $D_2$ . The height of the pole is  $h$ . The taper  $t$  of a pole is defined as  $(D_1 - D_2)/h$ . Figure 9(c) shows relative positions between a pole and a target object. Two distances  $d_1$  and  $d_2$ , and two heights  $h_1$  and  $h_2$  are added to feature values.

#### 5. FEATURE VALUES OF 2D IMAGE

##### 5.1 Cropping Images of Target Objects

Normally, multiple cameras are mounted on the MMS, and many images are taken while the vehicle is moving. With the MMS we used in this research, images were taken every 0.1 seconds. Since the same object appears in some images, it is necessary to select the image most suitable for object recognition.

The MMS records the time of acquisition of points and images. When a target object is extracted from a point-cloud, the average time  $\bar{t}$  is calculated from points of the target object. Images

captured during  $\bar{t} \pm \Delta t$  are selected as candidates. In this research, we set  $\Delta t = 2$  sec. Then, the points of a target object are projected onto each image, and the region of the target object is determined on the image. The image is selected according to the following criteria.

- (1) All points of a target object are projected on the image.
- (2) The number of pixels in the projected region is the largest.

Figure 10 shows three images that were sequentially captured. In these images, the image in Figure 10(c) satisfies these criteria.

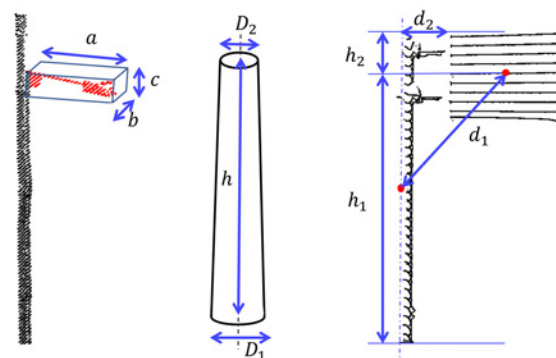
When points are projected onto an image, the pinhole camera model is used. For projection, the camera parameters, the relative positions between the laser scanner and digital cameras are required. By using these parameters, points can be projected on a digital image, as shown in Figure 11. In this research, we use the pinhole camera model proposed by Zhang, et al. (Zhang, 2000).

When the resolution of the selected image is larger than the resolution of a point-cloud, projected points cannot fill the region of the image. Therefore, we create the bounding box of the projected points, and crop the rectangle region from the image. In Figure 12 shows an image, the projected points, and the cropped image.

The cropped images are used for object recognition. To input them into CNN, the sizes of images are regularized to  $227 \times 227$  pixels, which is the required size for CNN we used in this research. Since the selected region is not a square in general, black pixels are added at the boundary of the region, as shown in Figure 12(d).

Table 1. Feature values of the point-cloud of a pole-like object

	Feature Values
Target Object	Width ( $a$ ), depth( $b$ ) and height ( $c$ ) of bounding box
	Eigenvalues ( $\lambda_1, \lambda_2, \lambda_3$ ) of PCA
	Ratios of eigenvalues ( $r_1, r_2, r_3$ )
	Ratios ( $r_l, r_p, r_s$ ) of linear, planar, and spatial points
	Angle ( $\theta$ ) of the first PCA direction from the ground
	Average of intensity values ( $I$ )
Pole	Height ( $h$ )
	Taper of cone ( $t$ )
	Diameters ( $D_1, D_2$ ) of the bottom and the top circle
Relative Position	Distance ( $d_1$ ) of the center of the target object from the pole axis
	The minimum distance ( $d_2$ ) of the target object from the pole axis
	Height ( $h_1$ ) of the target object from the bottom
	Height ( $h_2$ ) of the target object from the top



(a) Bounding box (b) Pole sizes (c) Relative positions  
Figure 9. Feature values of poles and target objects

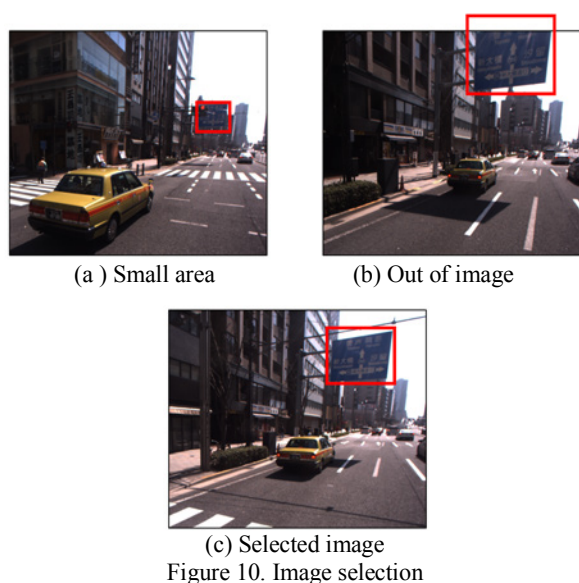


Figure 10. Image selection

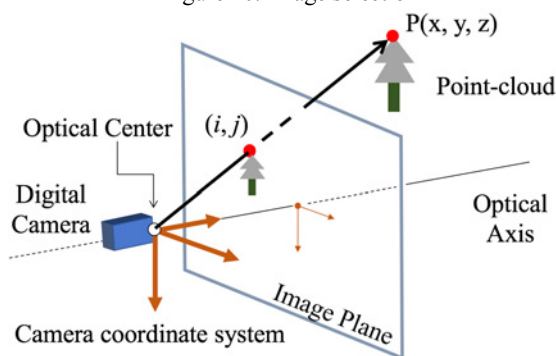


Figure 11. Projection of 3D points on an image

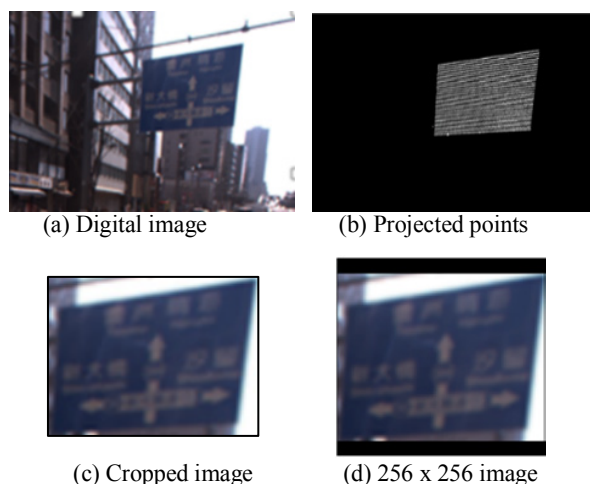


Figure 12. Cropping the image of a target object

## 5.2 Feature Values of Images

For obtaining feature values from images, we generate an image with  $227 \times 227$  pixels for each target object. Then feature values are automatically calculated using CNN. In our setting, a 4096-dimensional vector is output as feature values for each target object.

CNN calculates feature values on multiple layers with different image resolutions. In this research, we use AlexNet that has learned ImageNet. AlexNet is a convolutional neural network,

and ImageNet is a large scale learning data set of images (Krizhevsky, et al., 2012). AlexNet has five layers for convolution, normalization, and pooling. Convolution is the image filtering operation. Normalization is the operation to reduce the influence of brightness and contrast of the input image. Pooling is the operation to lower the resolution by setting the maximum value within a certain range of pixels.

Following the five layers, AlexNet has three fully connected layers, each of which outputs a vector of feature values. In this research, we use 4096 feature values generated from the second fully connected layer.

## 5.3 Selection of Important Features

The dimension of feature vectors for an image is much larger than the dimension of ones for a point-cloud. Furthermore, 4096-dimensional vectors are very sparse and the most elements are 0.

Therefore, we reduce the dimension of feature values by applying the random forest method to a set of 4096-dimensional vectors. The random forest allows us to calculate the importance of each feature. Roughly speaking, the importance of a feature represents how the feature is needed to classify the data set. The dimension of feature values can be reduced by selecting features with high importance values.

In this research, we reduce the dimension to 64 feature values. When a set of images of target objects are given as learning data, 4096 feature values are calculated for each target object using CNN. Then the random forest method is applied for classification, and the features with the top 64 importance values are selected.

## 5.4 Classification of Pole-like Objects

For each target object, 22 feature values are calculated from 3D points, and 64 features are calculated from an image. These feature values are combined into a feature vector with 86 values. In this research, we classify pole-like objects using the random forest, because the random forest is robust to outliers.

# 6. EXPERIMENTAL RESULT

## 6.1 Data and Method

Point-clouds and digital images for evaluating our method were captured in residential districts in Japan. In our experiment, we automatically extracted pole-like objects and target objects from point-clouds. The image of each target object was also automatically cropped from candidate images. Then we visually identified the class of each pole-like object, and manually classified the pole-like objects into the predefined classes.

We classified pole-like objects into destination signs, pedestrian signals, traffic signals, street lights, and traffic signs, as shown in Figure 13. We introduced the exceptional class for pole-like objects that are not included in the predefined classes. Pole-like objects in the exceptional class includes commercial signs, boxes, trees, and so on. Table 2 shows the numbers of pole-like objects in our dataset. For evaluation, we randomly divided the dataset into two halves, and made them into learning data and test data.

For comparison, we classified objects using three methods. One is the point-based method, which classifies using only point-clouds. The second is the image-based method, which uses only images. The third is the integrated method, which uses both

point-clouds and images. We used CNN for extracting feature values of images, and used the random forest for classifying pole-like objects.

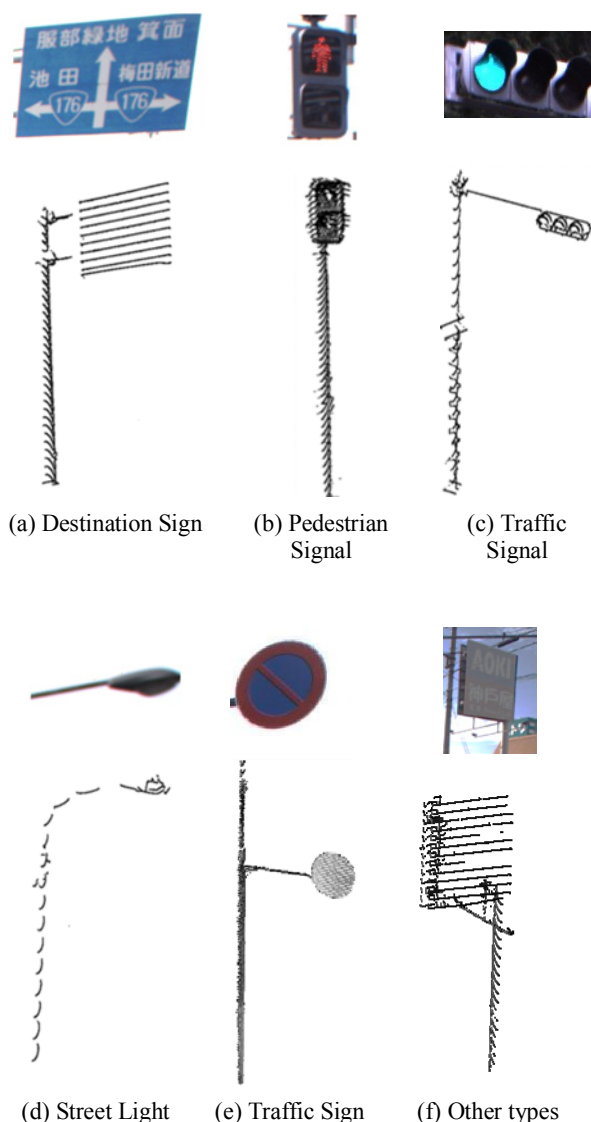


Figure 13. Type of pole-like objects

Table 2. Number of pole-like objects

Types	Number
Destination Sign	60
Pedestrian Signal	56
Traffic Signal	136
Street Light	204
Traffic Sign	82
Exceptional Objects	290

Table 3. Recognition rate only using points.

	Desti- nation Sign	Pedest- rian Signal	Traffic Signal	Street Light	Traffic Sign	Excep- tion
Destination Sign	28	0	0	0	0	2
Pedestrian Signal	0	25	0	0	0	3
Traffic Signal	0	0	53	2	0	14
Street Light	0	0	0	100	0	2
Traffic Sign	0	2	0	0	37	2
Others	0	1	5	9	3	127
Recall	93.3%	89.3%	76.8%	98.0%	90.2%	87.6%
Precision	100%	89.3%	91.4%	90.1%	92.5%	84.7%
F-measure	96.6%	89.3%	83.5%	93.9%	91.4%	89.9%

Table 4. Recognition rate only using images.

	Desti- nation Sign	Pedest- rian Signal	Traffic Signal	Street Light	Traffic Sign	Excep- tion
Destination Sign	25	0	0	1	4	4
Pedestrian Signal	0	26	0	0	0	2
Traffic Signal	0	0	52	10	1	6
Street Light	0	0	2	97	0	3
Traffic Sign	0	2	5	1	31	2
Others	0	0	0	2	2	141
Recall	83.3%	92.9%	75.4%	95.1%	75.6%	97.2%
Precision	100%	92.9%	88.1%	87.4%	91.2%	89.2%
F-measure	90.9%	92.9%	81.2%	91.1%	82.7%	93.1%

Table 5. Recognition rate using points and images.

	Desti- nation Sign	Pedest- rian Signal	Traffic Signal	Street Light	Traffic Sign	Excep- tion
Destination Sign	29	0	0	0	0	1
Pedestrian Signal	0	26	0	0	0	2
Traffic Signal	0	0	57	0	0	10
Street Light	0	0	0	102	0	0
Traffic Sign	0	0	0	0	36	2
Others	0	0	0	7	0	138
Recall	96.6%	92.8%	83.3%	100%	87.8%	95.1%
Precision	100%	92.8%	98.3%	92.7%	100%	90.2%
F-measure	98.3%	92.9%	90.5%	96.2%	93.5%	93.7%



## 6.2 Results

Table 3-5 show experimental results of classification for three methods. We calculated recall, precision, and F-measure. F-measure is the harmonic mean of recall and precision. Table 6 summarizes these results.

Comparing the F-measure scores of the point-based method and the image-based method, the point-based method was superior in recognizing destination signs, traffic signs, street lights, and traffic signs. On the other hand, the image-based method was superior in recognition of pedestrian signals and exceptional objects. This result indicates that point-clouds and images are complementary for classifying pole-like objects.

In the integrated method, the F-measure scores were improved in all classes, as shown in Table 6. Figure 14 shows the important values of the top 10 most important features. In this figure,  $P$ : means a feature from point-clouds, and  $I$ : means a feature from images. In this table, variables of point-clouds are defined in Table 1, and the number of the image feature indicates the sequential number in 4096 features. In the top 10 features, 6 features are from point-clouds and 4 features are from images. This result also shows that point-clouds and images are complementary for classifying pole-like objects.

The integrated method incorrectly classified several cases when the point-based method and the image-based method output different results. Figure 15 shows an example. In this case, the pedestrian signal without light was classified into the exceptional class, although the point-based method correctly recognized it as a pedestrian signal. This is because the image of the pedestrian signal without light is rare and it is similar to the image of a switchboard, which was learned as the exceptional class. However, we believe that this problem will be solved if the system learns many signals without light.

Table 6. F-measure of classified objects

	Desti- nation Sign	Pedest- rian Signal	Traffic Signal	Street Light	Traffic Sign	Excep- tion	Total
Points	96.6%	89.3%	83.5%	93.9%	91.4%	89.9%	90.6%
Images	90.9%	92.9%	81.2%	91.1%	82.7%	93.1%	88.4%
Points and Images	98.3%	92.9%	90.5%	96.2%	93.5%	93.5%	93.7%

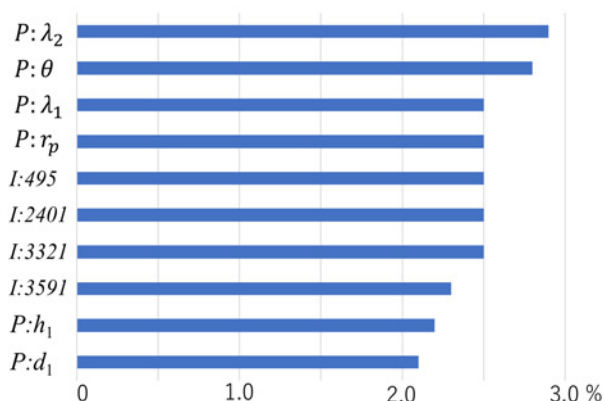
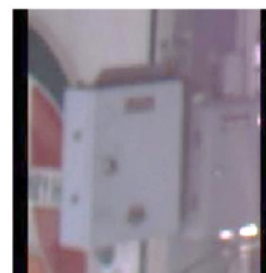


Figure 14: Importance of feature values.



(a) Pedestrian Signal



(b) Switchboard

Figure 15: An example incorrectly classified.

## 7. CONCLUSION

In this paper, we proposed a classification method for pole-like objects. In our method, poles and target objects were extracted from point-clouds, and feature values were calculated using point-clouds. Then points of each target object were projected onto an image, and the object region was cropped from the image. Feature values of the cropped image were calculated using CNN. The feature values of point-clouds and images are unified, and they are used for machine learning. For evaluating our method, we classified objects using three methods. The first method used only point-clouds, the second used only images, and the third used both point-clouds and images. The experimental results showed that the third method could most accurately classify pole-like objects.

In future work, we would like to improve the method for extracting target objects. We also would like to investigate how many classes can be identified using our method.

## ACKNOWLEDGEMENTS

MMS data in this paper are courtesy of AISAN Technology Co.Ltd. We would like to thank for their helpful support.

## REFERENCES

- Bolles, R., Fischler, M., 1981. A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data, *International Conference on Artificial Intelligence*, B.C., Canada, pp. 637-643.
- Breiman, L., 2001. Random Forest, *Machine Learning*, 45, pp. 5-23.
- Fukano, K., Hiroshi, M., 2015. Detection and Classification of Pole-Like Objects from Mobile Mapping Data, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. II-3/W5, pp.57-64.

- Golovinskiy, A., Kim, V., Funkhouser, T., 2009. Shape-Based Recognition of 3D Point Clouds in Urban Environments, *International Conference on Computer Vision*, Kyoto, Japan, pp. 2154-2161.
- Krizhevsky, A., Sutskever, I., Hinton G., 2012. ImageNet classification with deep convolutional neural networks, *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, Vol. 1, pp. 1097-1105.
- Mitsubishi Electric, 2018. Mobile mapping system high-accuracy GPS mobile measuring equipment. <http://www.mitsubishi-electric.com/bu/mms/catalog/pdf/catalog.pdf> (2 Apr. 2018)
- Soilán, M., Riveiro, B., Martínez-Sánchez, J., Arias, P., 2016. Traffic Sign detection in MLS acquired point clouds for geometric and image-based semantic inventory, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 114, pp. 92-101.
- Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, pp. 181-188.
- Zhu, X., Zhao, H., Liu, Y., Zhao, Y., Zha, H., 2010. Segmentation and classification of range image from an intelligent vehicle in urban environment, *The 2010 IEEE/RSJ International Conference*, Taipei, Taiwan, pp.1457-1462.
- Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., Liu, W., 2016a. Traffic Sign detection and recognition using fully convolutional network guided proposals, *Neurocomputing*, Vol. 214, pp. 758-766.
- Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S., 2016b. Traffic-Sign Detection and Classification in the Wild, *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2110-2118.