CLASSIFICATION FOR ROADSIDE OBJECTS BASED ON SIMULATED LASER SCANNING

Kenta Fukano[1], and Hiroshi Masuda[2]

1) Graduate student, Department of Intelligence Mechanical Engineering, The University of Electro-Communications, Tokyo, Japan.
2) Prof, Department of Intelligence Mechanical Engineering, The University of Electro-Communications, Tokyo, Japan.

Abstract: The maintenance of roadside objects, such as utility poles, traffic signs and streetlights, is very important for sustaining infrastructures, but it is costly and tedious work to survey a huge number of roadside objects. A mobile mapping system (MMS) is promising for improving the efficiency of maintenance tasks. In our previous work, we proposed a machine learning approach for automatically identifying roadside objects from point-clouds captured by a MMS. However, supervised machine-learning methods require training data, which have to be carefully created by manually classifying roadside features. In this research, we propose a method for automatically generating training data using 3D CAD models. Point-clouds of roadside objects are created from CAD models by simulating laser scanning on a moving vehicle. In our experiments, our method could generate reasonable point-clouds that can be used as training data for supervised machine learning.

Keywords: point-cloud, mobile mapping system, machine learning, classification

1. INTRODUCTION

The maintenance of roadside objects, such as utility poles, traffic signs and streetlights, is very important for sustaining infrastructures. However, it is costly and tedious work to survey a huge number of roadside objects. A mobile mapping system (MMS) is promising for improving the efficiency of maintenance tasks for roadside objects.

A MMS can capture point-clouds of road surfaces, buildings and roadside objects while running on a road. Since a various classes of objects are included in a point-cloud, it is necessary to segment a point-cloud into each object and to identify the object class (Golvinsky et al., 2009).

So far some researchers have studied methods for classifying objects in point-clouds captured by MMSs (Yokoyama et al., 2013; Cabo et al., 2014; Yang et al., 2015). They segmented point-clouds and identified object classes on roadsides. Their methods are based on threshold values of feature values, which are carefully determined by experiments. For classifying new objects, new thresholds have to be introduced. In addition, algorithms become complicated when many objects have to be classified.

Other researchers introduced supervised machine learning methods for classifying roadside objects (Golvinsky et al., 2009; Zhu et al., 2010; Ishikawa et al., 2013; Fukano et al., 2014). In supervised machine learning, threshold values are automatically determined using training data. In our previous work, pole-like objects were classified at high recognition rates using the random forest method.

However, it is time-consuming and tedious work for users to generate training data, because the users have to search for roadside objects in huge point-clouds and manually add labels to segmented point-clouds. Therefore, it is very useful to automatically generate training data for machine learning. Koenig and Howard (2004) proposed a method for generating training data for point-clouds by simulating terrestrial laser scanning (TLS). Gschwandtner et al. (2011) also proposed a simulation method for a laser scanner on a mobile robot. However, their methods are not suitable for point-clouds captured by a MMS, because their laser scanning methods are different from laser scanning on a MMS.

In this paper, we propose a method to automatically generate training data for MMS data. In our method, roadside objects are represented as 3D CAD models, and virtual point-clouds are generated by simulation of laser scanning on a moving vehicle. Noises on point-clouds are also simulated. Then we classify roadside objects using virtual training data, and compare recognition rates to actual training data.

2. SIMULATION OF LASER SCANNING ON VIECLES

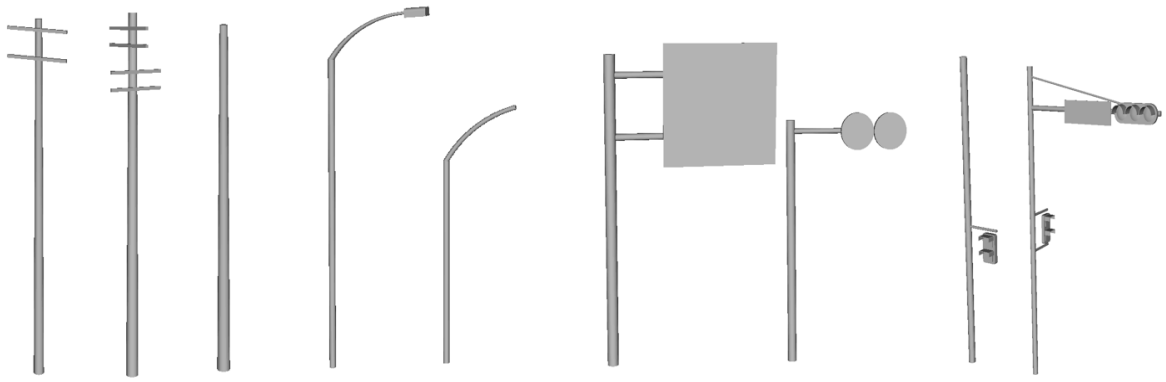
2.1 3D Models of Roadside Objects

We consider classifying utility poles, streetlights, traffic signs and signals, which are typically located on roadsides. These roadside objects have common shapes in most cases. Especially, shapes of utility poles, streetlights on highways, traffic signs, and signals are determined as the standards in Japan. Although there are no common standards for streetlights in residential areas, we can recognize typical shapes of streetlights by observing point-clouds.

When standard shapes of objects are given, their 3D models can be created using a shape modeling system, such as 3D CAD. We created 3D models of utility poles, streetlights on highways, traffic signs, and signals based on the standards. We also examined typical shapes of streetlights in residential areas and created their 3D models.

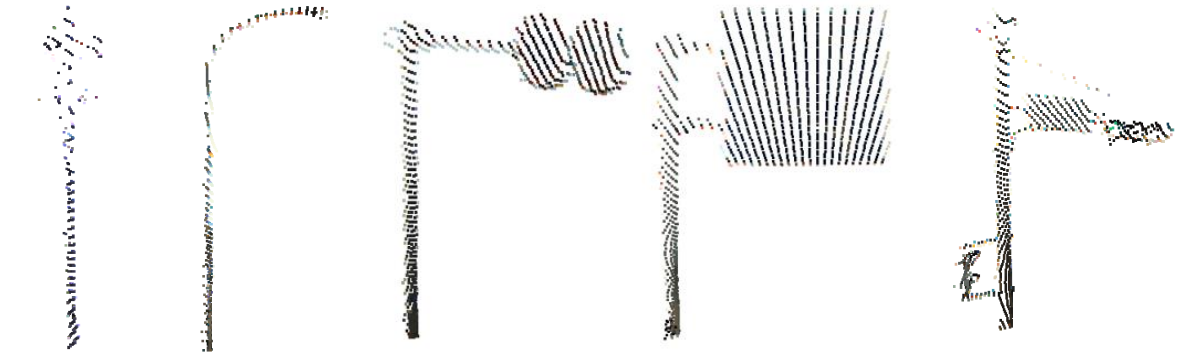
CAD models are converted to triangular meshes in the STL format. Figure 1 shows triangular mesh models of utility poles, streetlights, traffic signs and signals.

Figure 2 shows actual point-clouds of roadside objects. As shown in Figure 1 and 2, 3D models are similar to actual point-clouds, but they do not have the exactly same shapes. Actual roadside objects may have various forms of attachments, such as cables and signboards. However, we modeled only common bodies of roadside objects and disregarded detail attachments, because there is a wide range of variations in attachments.



Utility pole Streetlight Traffic sign Signal

Figure 1. Created mesh models



Utility pole Streetlight Traffic sign Signal

Figure 2. Actual point-clouds

2.2 Simulation of Laser Scanning on A Moving Vehicle

Point-clouds of roadside objects can be generated using 3D models. First we consider laser scanning for a MMS. The trajectory of a laser beam emitted from a laser scanner is represented as a straight line, which is determined using the scanner position and the laser beam direction. Straight lines can be calculated using the scanner position on a vehicle, the speed of a vehicle, and the scan range, resolution and frequency of a laser scanner. In our experiments, we used "SICK LMS 291" as laser scanners on a MMS. The specification is shown in Table 1. The laser scanners are tilted at 30 [degree], and capture points within 180 [degree], as shown in Figure 3(b).

Table 1. Specification of SICK LMS 291

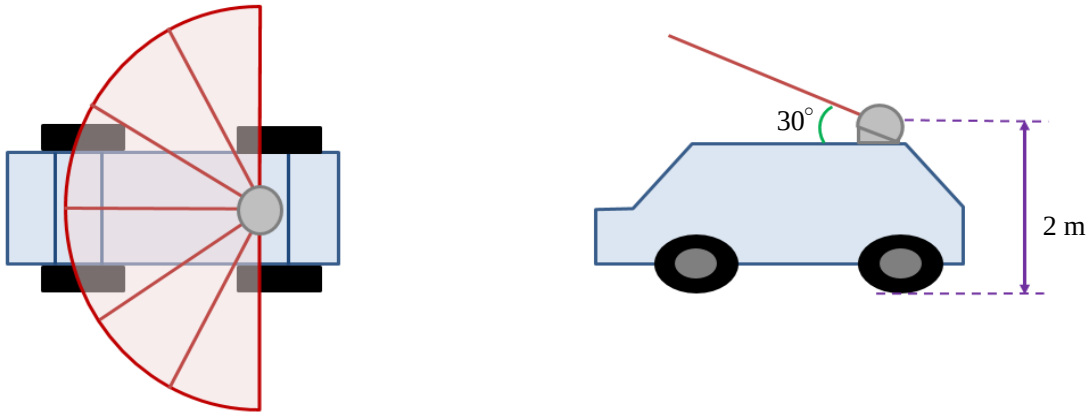
Parameters	Value
Scan Range (degree)	180
Resolution (pts/scan)	180
Frequency (Hz)	37.5

We describe T =1/37.5 [sec] as the time of a single scan, t =T/180 [sec/pts] as a time step, v =11.11 [m/s] as a vehicle speed, and $\boldsymbol{D}$ as a vehicle direction. When the initial position of the laser scanner is $\boldsymbol{P}_0$, it moved to $\boldsymbol{P}_n = \boldsymbol{P}_0 + nvt\boldsymbol{D}$ after n time steps. Then the straight line of a laser beam can be represented as $L_n : \boldsymbol{x} = \boldsymbol{P}_n + k\boldsymbol{V}_n$, where $\boldsymbol{V}_n$ is the laser beam direction after n time step, and k is a parameter of a straight line.

Figure 4 shows roads, roadside objects, and a vehicle that were used in our simulation. Roads have two

lanes, and the vehicle runs on the left lane in both directions. We determined the width of roads and positions of roadside objects, as shown in Figure 4. Plates of traffic signs are placed in directions perpendicular to vehicle directions.

Point-clouds can be created by calculating intersection points between lines L_n and a triangular mesh, as shown in Figure 5. When a straight line intersects with multiple triangles, only the nearest point is remained. Simulated point-clouds are shown in Figure 6.



(a) Top view (b) Side view

Figure 3. Outline figure of a simulated MMS

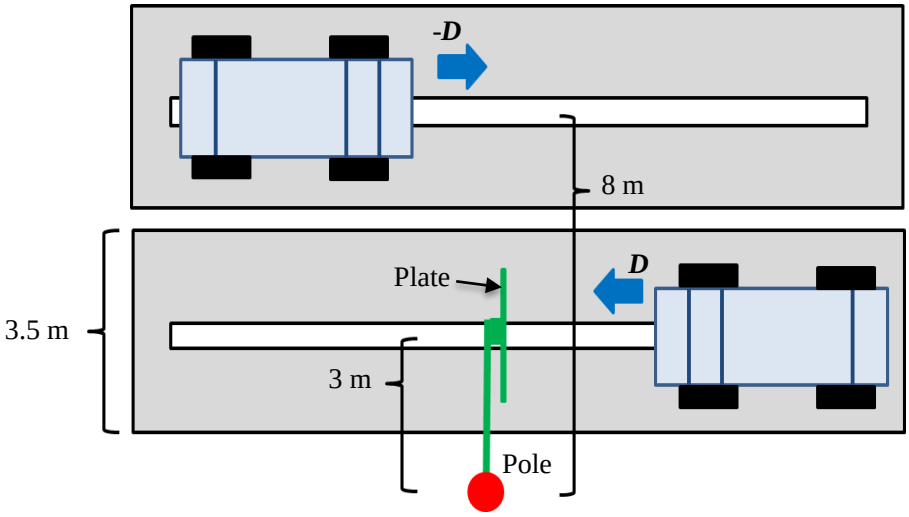


Figure 4. Model of a vehicle moving on the road

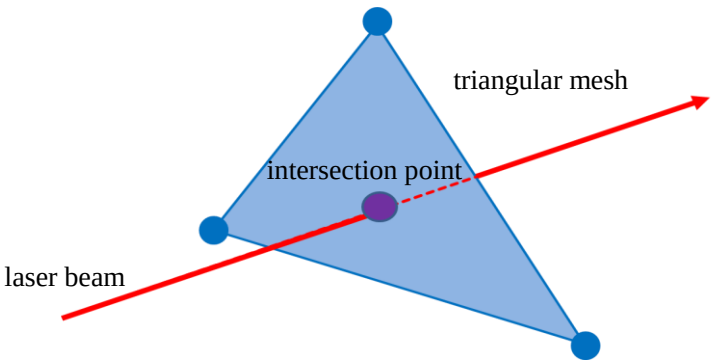


Figure 5. Generation of a point

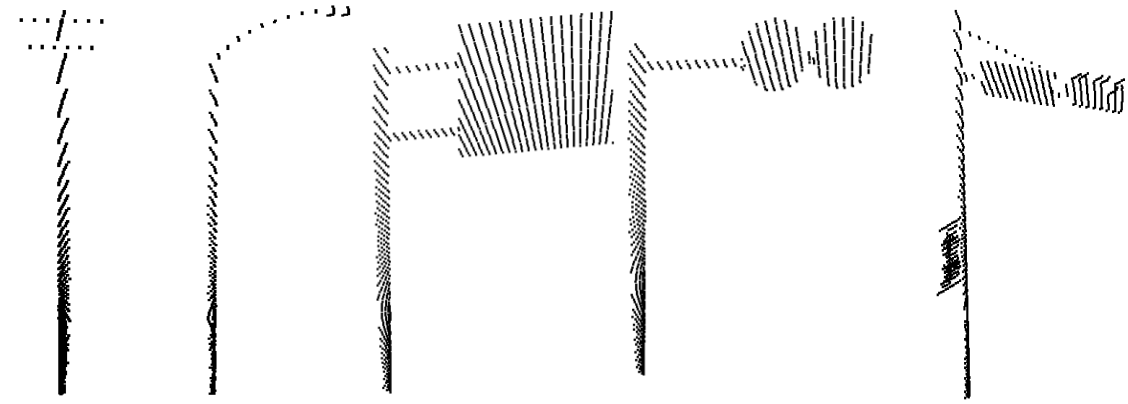


Figure 6. Simulated point-clouds

3. CLASSIFICATION

3.1 Feature Values

Machine learning techniques classify objects using feature values, which are calculated from point-clouds of objects. Typical feature values for point-clouds are geometric feature values, such as eigenvalues of the principal component analysis (PCA) (Ishikawa et al., 2013; Yokoyama et al., 2013; Zhu et al., 2010; Golvinsky et al., 2009).

Pole-like objects are extracted as subsets of a point-cloud using the method proposed by Masuda et al (2013). Then the following feature values are calculated using points of each pole-like object.

(1) Size of a bounding box of a point-cloud: (w, d, h)
(2) PCA eigenvalues of a point-cloud: (λ_1, λ_2, λ_3)
(3) Ratios of PCA eigenvalues: (λ_1/λ_2, λ_2/λ_3, λ_1/λ_3)
(4) Distribution of the edge direction: (r_1, r_2, r_3)

For calculating the sizes (w, d, h) of a bounding box, three axes $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ of a bounding box are determined so that $\boldsymbol{k}$ is the z-axis, and $\boldsymbol{i}$ and $\boldsymbol{j}$ are eigenvectors of PCA of (x, y) coordinates. For calculating the distribution of the edge direction, we create a k-neighbor graph of points by generating edges between neighbor points. Then, we smooth the edges with the Taubin filter (Taubin, 1995), which is a low-pass filter for 3D shapes. The smoothed results are shown in Figure 7. We classify edges in a k-neighbor graph into three groups based on their angles from a horizontal plane. If the angle is larger than 60 degree, it is classified into the group of vertical edges. If the angle is less than 30 degree, it is classified into the group of horizontal edges. Other edges are classified into the group of angled edges. We count the number of edges of each group. Finally, we calculate ratios of the number of edges as (r_1, r_2, r_3). These values are effective to distinguish trees from man-made objects (Fukano et al., 2014).

3.2 Subdivision of Points into Five Subsets

Since pole-like objects tend to have similar feature values, it is difficult for a machine learning method to classify them in a high recognition rate. Therefore we divide points of an object into three components and calculate the feature values of each subset. As shown in Figure 8, cylinders and planes are extracted from points, and five point-clouds A, B, C, D, and E are generated. When multiple cylinders or planes are detected, only the largest one is selected. Feature values (w, d, h), (λ_1, λ_2, λ_3), and (λ_1/λ_2, λ_2/λ_3, λ_1/λ_3) are calculated using five point-clouds. Distribution (r_1, r_2, r_3) are calculated only from all points A.

In addition, we introduce two more feature values for describing relationships between subsets, as follows. Distances are calculated using the center of each subset of points. d_c is the distance between B and C, and d_p is the distance between D and E.

(5) Distance between two components: (d_c, d_p)
(6) Number of points in each component: (n_A, n_B, n_C, n_D, n_E)

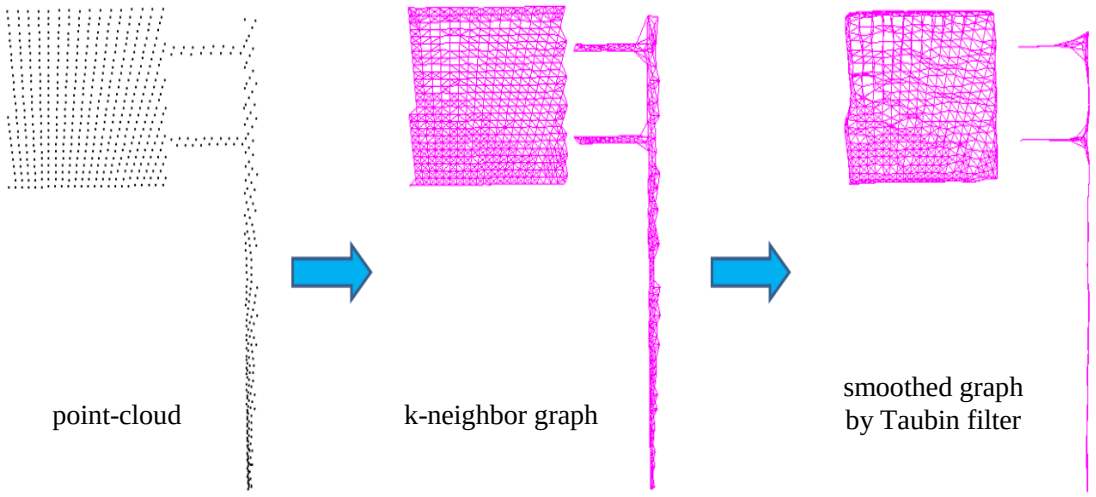


Figure 7. Smoothing k-neighbor graph with Taubin filter

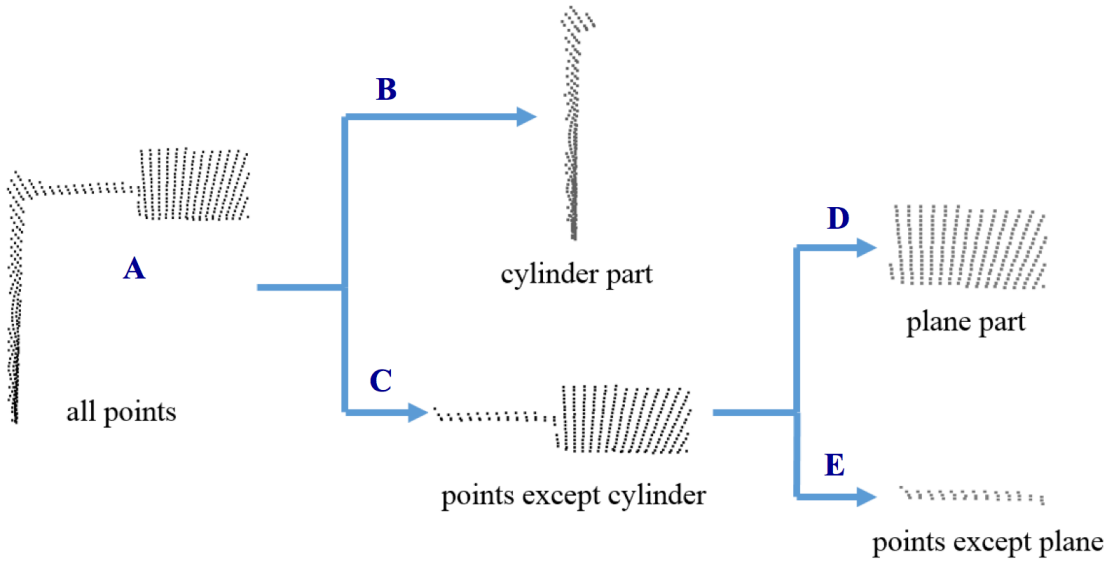


Figure 8. Five subsets of points

3.4 Classification Result

We classified objects using training data based on virtual points, which were generated using 3D models in Figure 1. We also classified objects using training data based on actual points, which were generated by manually adding labels to points of pole-like objects. The numbers of training data are 50 for both cases. The classes of objects were utility poles, streetlights, traffic signs, and traffic signals. The numbers of objects were shown in Table 2.

We applied the random forest using training data and evaluated recognition rates. The recognition rates were measured using F-measure, which is a harmonic mean of precision and recall. As shown in Table 2, the recognition rates based on simulated points were reasonable, but they were considerably lower than the ones of actual points.

Table 2. Recognition rates (F-measure)

Objects	Number of objects	Virtual point-clouds for training data	Actual point-clouds for training data
Utility pole	360	87.8 %	93.4 %
Streetlight	243	80.8 %	93.7 %
Traffic sign	144	79.6 %	96.5 %
Signal	108	56.7 %	79.0 %

4. SIMULATION OF NOISY POINT-CLOUDS

4.1 Simulation of Noisy Points

In our observation, the distribution of actual points is more scattered than one of simulated points. Therefore, we add noises to coordinates of simulated point-clouds. In laser scanning, the directions of laser beams are controlled mechanically, and the distances are measured by the round-trip time of laser beams. Since distance values are more sensitive to measurement conditions such as object colors and materials, we add noises only to distance values of simulated points.

When an intersection point is calculated between a triangle and a laser beam, the coordinate is perturbed in the forward or backward direction with equal probability, as shown in Figure 9.

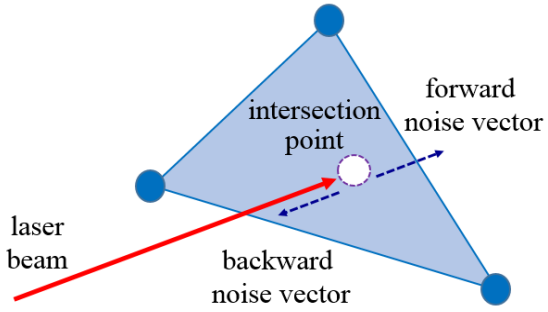


Figure 9. Direction of noises

In existing methods for simulating laser scanning, the distribution of noise is calculated according to a distance between a scanner and an object (Gschwandtner et al., 2011; Koenig & Howard, 2004). However, in the case of MMSs, noises are caused by positioning of a vehicle, calibration of a laser scanner on a vehicle, and relative distances from a laser scanner.

To identify dominant factors of noises, we investigated the distribution of noises of actual points. We extracted planar parts from point-clouds, and calculated plane equations using the principal component analysis. Then we measured residual errors of points on the planes. When we calculated a correlation between residual errors and heights from the ground, the correlation coefficient was 0.0588, which is a small value. This result shows that noises are not sensitive to object distances and positioning errors of a scanner are more dominant for coordinates of points.

The distribution of residual errors is shown with the normal distribution in Figure 10. We generated the histogram of residual errors on planes and calculated the probability distribution of residual errors. The standard deviation was 5.67 mm. In Figure 10, the normal distribution $N(\mu, \sigma^2)$ is shown with $\mu = 0$ and $\sigma = 0.567$. We can observe that the distribution of residual errors do not exactly follow the normal distribution. Therefore, we generated noise values that followed the probability distribution of actual points instead of adding normal distribution noises.

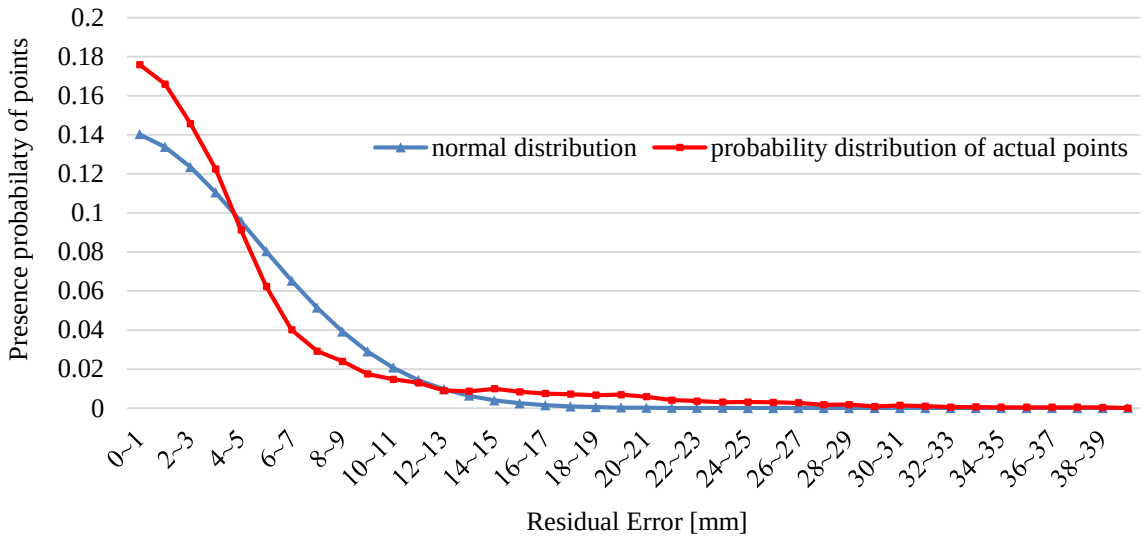


Figure 10. Probability distribution and normal distribution

4.2 Experimental Results

We classified roadside objects using training data based on simulated points. We also classified the same objects using the training data based on actual points, and compared the recognition rates. The result is shown in Table 3.

In our experiments, the recognition rate based on simulated noise-free points was considerably lower than the one based on actual point-clouds. When noises were added to simulated points, the recognition rate was improved for all objects. Especially, that recognition rate of signals increased from 56.7% to 72.0%. The average recognition rate was improved from 76.2% to 84.4%. The result shows that simulated points with noises are effective as training data for supervised machine learning.

In our experiments, the recognition rate of signals was worse than those of other objects in both cases of simulated and actual points. This is because some utility poles were classified into signals. Figure 11 (a) shows examples of utility poles classified into signals. In these examples, utility poles have multiple attachments, such as cross arms, plates and power lines. On the other hand, some signals have two traffic lights for cars and pedestrians, as shown in Figure 11 (b). In both objects, three PCA eigenvalues (λ_1, λ_2, λ_3) are relatively large for subsets C in Figure 8, and they are not good discriminators for objects in Figure 11 (a) and (b). We consider other feature values are required to improve recognition rates of signals.

Table 3. Recognition rate

Objects	Simulated points *without* noises	Simulated points *with* noises	Actual points of objects
Utility pole	87.8 %	91.2 %	93.4 %
Streetlight	80.8 %	86.1 %	93.7 %
Traffic sign	79.6 %	88.6 %	96.5 %
Signal	56.7 %	72.0 %	79.0 %
Average	76.2%	84.4 %	90.6 %

(a) Utility poles recognized as signals (b) Pole with two signals

Figure 11. Points classified into signals

5. Conclusion

We proposed a method for automatically generating training data using 3D CAD models. In our method, points of roadside objects are created from CAD models by simulating laser scanning of a moving vehicle. We first generated noise-free point-clouds and then added noises that follow the probability distribution measured from actual point-clouds. The results showed that object classification using simulated noisy point-clouds as training data achieved reasonable recognition rates.

Recognition rates could be improved by the addition of noise, but they are still lower than the ones of actual point-clouds. In future works, we would like to improve recognition rates by investigating differences between simulated points and actual points. We also would like to classify roadside objects into more subdivided classes.

ACKNOWLEDGEMENTS

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

REFERENCES

Golovinskiy, A., Kim, V. G. and Funkhouser, T. (2009). Shape-based Recognition of 3D Point Clouds in Urban Environments, *IEEE 12th International Conference*, Kyoto, Japan, pp. 2154-2161.

Cabo, C., Ordonez, C., Garcia-Cortes, S. and Martinez, J (2014). An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, pp. 47–56.

Yang, B., Dong, Z., Zhao, G., Dai, W. (2015). Hierarchical extraction of urban objects from mobile laser scanning data, *ISPRS Journal of Photogrammetry and Remote Sensing*, 99, pp. 45–57.

Fukano, K., Oguri, S. and Masuda, H. (2014). Recognition and Reconstruction of Roadside Objects Based on Mobile Mapping Data, *15th International Conference on Precision Engineering (ICPE)*,

Ishikawa, K., Tonomura, F., Amano, Y. and Hashizume, T. (2013). Recognition of Road Objects from 3D Mobile Mapping Data, *International Journal of CAD/CAM*, 13, pp.41-48.

Yokoyama, H., Date, H., Kanai, S. and Takeda, H. (2013). Detection and Classification of Pole-like Objects from Mobile Laser Scanning Data of Urban Environments, *International Journal of CAD/CAM*, 13, pp.31-40.

Taubin, G. (1995). A Signal Processing Approach to Fair Surface Design, *Proceeding of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 351-358.

Gschwandtner, M., Kwitt, R., Uhl, A. and Pree, W. (2011). BlenSor: blender sensor simulation toolbox. *Advances in Visual Computing*, pp. 199-208.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator, *Proceeding of 2004 IEEE/RSJ on Intelligent Robots and Systems (IROS 2004)*, 3, pp. 2149-2154.

Masuda, H., Oguri, S. and He, J. (2013). Shape Reconstruction of Poles and Plates from Vehicle- Based Laser Scanning Data, *Proceeding of Informational Symposium on Mobile Mapping Technology*, CD - ROM.