

POLE-LIKE OBJECTS RECOGNITION FROM MOBILE LASER SCANNING DATA USING SMOOTHING AND PRINCIPAL COMPONENT ANALYSIS

H. Yokoyama ^a, H. Date ^a, S. Kanai ^a, H. Takeda ^b

^a Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan

h_yokoyama@sdm.ssi.ist.hokudai.ac.jp, (hdate, kanai)@ssi.ist.hokudai.ac.jp

^b Kokusai Kogyo Co., Ltd, Chiyoda-ku 102-0085, Japan, htake@kkc.co.jp

KEY WORDS: Mobile Laser Scanning, Object Recognition, Laplacian Smoothing, Point Clouds, Principal Component Analysis, Pole-like Objects

ABSTRACT:

With the spread of the Mobile Laser Scanning (MLS) system, the demands for the management of road and facilities using MLS point clouds have increased. Especially, pole-like objects such as streetlights, utility poles, street signs and etc. are in high demand as facilities to be managed. We propose a method for recognizing pole-like objects from MLS point clouds. Our method is based on Laplacian smoothing using the k-nearest neighbors graph, Principal Component Analysis for recognizing points on pole-like objects, and thresholding for the degree of pole-like objects. Our method can robustly recognize pole-like objects with various radii and tilt angles from MLS point clouds. For correctly segmented objects, accuracy of pole-like object recognition is on average 97.4%.

1. INTRODUCTION

With the development of inexpensive and high accuracy laser scanner devices, the Mobile Laser Scanning (MLS) system which installed these devices on a car has been widely used. The MLS point clouds can be useful not only for building and city modeling, but also for managing various facilities in urban environments. In particular, pole-like objects such as streetlights, utility poles, street signs, and etc. are in high demand as facilities to be managed, and it is required to recognize them from urban MLS point clouds. However, manually recognizing these pole-like objects from large point clouds data requires a great deal of time and cost. Therefore, for efficient management of facilities, it is necessary to automatically recognize pole-like objects from MLS point clouds.

Much research on pole-like objects recognition from MLS point clouds has been conducted. Existing methods are based on machine learning (Golovinskiy 2009b, Lai 2009), or the arrangement and position of measurement points (Manandhar 2001, Lehtomaki 2010). These methods have some problems, such as they require a lot of training data, or that they cannot recognize pole-like objects with different radii, and tilt angles.

In this paper, we propose an algorithm to automatically recognize pole-like objects from MLS point clouds. Our method is based on Laplacian smoothing using the k-nearest neighbors graph, Principal Component Analysis (PCA) for recognizing points on the pole-like objects, and thresholding for the degree of pole-like objects. By using smoothing and PCA, robust recognition of the pole-like objects with various radii and tilt angles is realized. Details of the proposed method are described in chapter 3.

2. RELATED WORKS

Existing research on automatically recognizing or classifying objects, including pole-like objects, are introduced in this chapter.

Based on machine learning, Golovinskiy et al. (Golovinskiy 2009b) propose a method to classify various objects such as cars, streetlights, trees, fire hydrant, and etc. from the combination of MLS and ALS point clouds. A major feature of their method is high precision segmentation by the graph cut algorithm. To classify objects, they input feature quantities of objects into the Support Vector Machine (SVM). As learning data, they used a part of the input data which are classified manually. In order to increase the recognition rate, a sufficient amount of learning data must be required. They also indicated that the better shape descriptors and classifiers are required for better results. As a similar method using machine learning, Lai et al. (Lai 2009) attempted to classify objects in MLS point clouds by using a lot of 3D data that exists on World Wide Web as learning data. In the learning method, sufficient training data based on input point clouds is necessary for suitable learning.

Based on the arrangement and position of measurement points, Manandhar et al. (Manandhar 2001) detected vertical poles from MLS point clouds. In their research, the MLS point clouds consisted of vertical scan lines and vertical poles by extracting vertical line segments from individual scan lines. Their detection method was limited to the extraction of the vertical poles. Due to this limit, the method cannot detect tilted poles and is not applicable to arbitrary point clouds. Lehtomaki et al. (Lehtomaki 2010) extracted sweeps that were expected to be measured pole-like objects from MLS point clouds. Then they found another sweep either below the current sweep or above, and made them a cluster. However the method cannot recognize pole-like objects with a specific radii and limited point densities.

In this paper, we propose a robust recognition method of the pole-like objects with arbitrary tilt angles and radii from MLS point clouds without learning data, by using smoothing and PCA.

3. POLE-LIKE OBJECTS RECOGNITION METHOD

3.1 Overview of the Method

The proposed algorithm is shown in Fig.1. In the method, we assume that ground points are already removed from given point clouds. The algorithm of our method consists of four phases. First, the input point clouds are segmented, as the result the points estimated on each object are grouped. Second, smoothing is applied to each segment. Third, each point is classified into the points on the pole-like objects, on the planar objects, and on other objects. Finally, the degree of the pole-like objects of each segment is evaluated, and the segments of the pole-like object are extracted by thresholding.

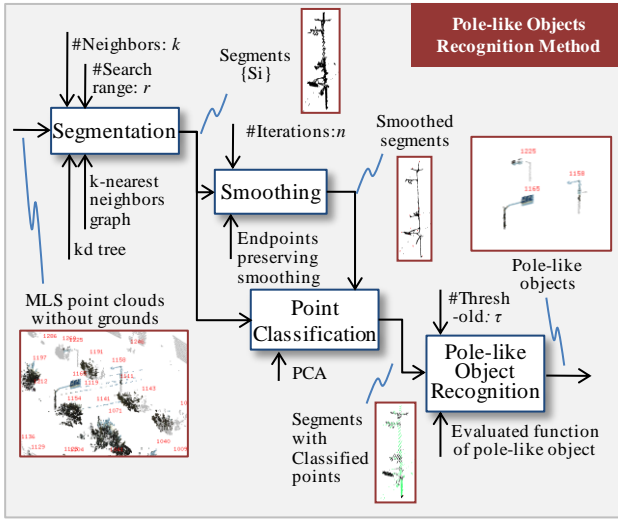


Figure 1. Proposed algorithm of pole-like objects recognition

3.2 Segmentation

Many segmentation techniques of point clouds have already existed (Golovinskiy 2009a, Moosmann 2009, Wang 2008). In our implementation, we adopt the simple segmentation method by connecting the nearest neighbor points. As a result, the segment is composed of the point set of the connected k-nearest neighbors graph. The graph is generated to sequentially generate an edge among the k-nearest neighbors between the point i and points included in a sphere with a radius r centered at point i . Even if the neighbors included in the sphere are less than k , we do not expand the radius r .

3.3 Endpoint Preserving Laplacian Smoothing

The robust extraction of points on pole-like objects by PCA described in next section is difficult because of the measurement noises, the bias of the point distributions, and the differences of the pole-like object radius. Therefore, to improve the recognition rate of the points on the pole-like objects by PCA, endpoint preserving Laplacian smoothing is applied to the k-nearest neighbors graph in our method.

Generally, smoothing is applied for the purpose of removing the noise of the measurement data. On the other hand, recently,

smoothing is used for other purposes. For example, some methods of skeleton extraction from point clouds have been proposed (Au 2008). The purpose of their research is to extract the skeletons of wire-objects, and the recognition of the objects is not focused on.

In our method, smoothing is applied to the MLS point clouds in order to improve the recognition rate of the points on the pole-like objects by the PCA and to distinguish points on the pole-like objects from the ones on the planar objects, and the others. We focus on the exaggeration and the degeneration of the object shape features based on Laplacian smoothing which is an operation that moves each point to the centres of the neighbours. Laplacian smoothing makes the pole-like objects into a thin pole shapes through shape degeneration. As the result, the point distributions of the pole-like objects come to be degenerated into a one dimensional distribution (Fig.2). In addition, measurement noises are removed. Therefore, applying the Laplacian smoothing to the scan data raises the recognition rate of the pole-like object points by the PCA. Laplacian smoothing is done by applying equation (1).

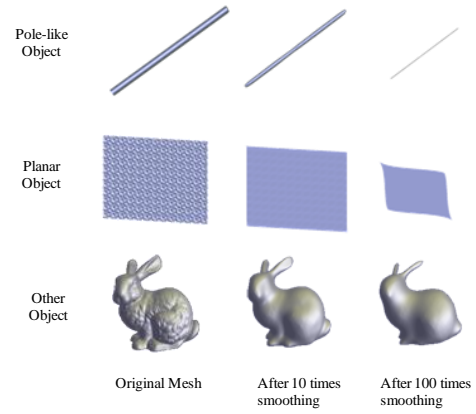


Figure 2. Examples of degenerated objects in mesh model by smoothing

$$\mathbf{p}'_i = \mathbf{p}_i + \lambda \Delta \mathbf{p}_i \quad (1)$$

where

\mathbf{p}'_i = position of point i after smoothing

\mathbf{p}_i = position of point i

λ = smoothing strength ($0 \leq \lambda \leq 1$)

$\Delta \mathbf{p}_i$ = Laplacian, and it is given by the equation (2)

$$\Delta \mathbf{p}_i = \sum_{j \in i^*} \omega_{ij} (\mathbf{p}_j - \mathbf{p}_i) \quad (2)$$

where

ω_{ij} = positive weight ($\sum_{j \in i^*} \omega_{ij} = 1$)

i^* = a set of the neighbors of point i

The point clouds of pole-like objects with various radii can be degenerated into a one dimensional distribution by iteratively applying equation (1), and then pole-like objects with various radii can be recognized.

However, Laplacian smoothing has the problem that the branching structures of the pole-like objects are lost. This causes the decrease of the recognition rate in the following object recognition. To solve this problem, we propose the endpoints preserving Laplacian smoothing, which controls

displacements during smoothing according to the distribution of the neighbors of a point. At the endpoint, the neighbors are distributed in one direction. On the other hand, the neighbors are distributed in all directions at the inner point. From this observation, the displacements during smoothing are controlled so as to preserve the endpoints according to the distribution of the neighbors. We evaluate whether point i is the endpoint or not using equation (3).

$$e(i) = \frac{1}{|V_i| \times \pi} \sum_{(j,k) \in V_i} \text{angle}(j,i,k) \quad (3)$$

where i^* = a set of neighbors of point i

$$V_i = \{(j,k) \mid j,k \in i^*, j \neq k\}$$

$$\text{angle}(j,i,k) = \text{angle of } j,i,k$$

If the point i is the endpoint, $\text{angle}(j,i,k)$ is small (Fig.3 left), then the $e(i)$ becomes small. On the other hand, if the point i is far from the endpoint, some $\text{angle}(j,i,k)$ are large (Fig.3 right), then $e(i)$ becomes large.

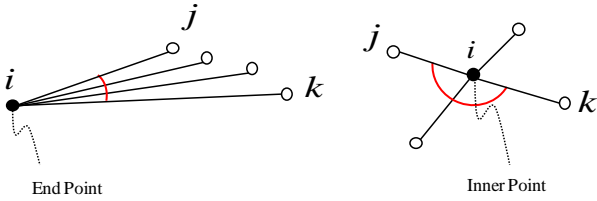


Figure 3. Neighbors' distribution at the endpoint and the inner point

Displacements in smoothing can be controlled by using the smoothing strength λ . Additionally in order to reduce influences of the distant points during smoothing, we use weight ω_{ij} which is in inverse proportion to the distance between points. Endpoints preserving Laplacian smoothing is done by the equations (4), (5), and (6).

$$\mathbf{p}_i' = (1 - \lambda)\mathbf{p}_i + \lambda \sum_{j \in i^*} \omega_{ij} \mathbf{p}_j \quad (4)$$

$$\lambda = e(i) \quad (5)$$

$$\omega_{ij} = \frac{|\mathbf{p}_j - \mathbf{p}_i|^{-1}}{\sum_{j \in i^*} |\mathbf{p}_j - \mathbf{p}_i|^{-1}} \quad (6)$$

Fig.4 shows the comparative result of the Laplacian smoothing and endpoint preserving Laplacian smoothing for the same utility pole. The number of iterations of the smoothing is 40. Fig.4(a) shows the input point clouds of the utility pole. Fig.4(b) and Fig.4(c) show the results of the Laplacian smoothing and endpoint preserving Laplacian smoothing respectively. After the endpoint preserving Laplacian smoothing, the branching structures still remain. During iterating smoothing, the distribution of the points on the pole-like objects with various radii becomes one dimension.

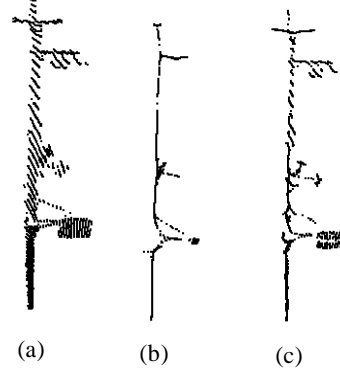


Figure 4. Comparative result of smoothing

(a) input point clouds of the utility pole

(b) afret Laplacian smoothing

(c) after endpoint preserving Laplacian smoothing (iteration times $n = 40$)

3.4 Point Classification

Each point is classified into three types which are the points on the pole-like object, the points on the planar object, and the others. The local point distributions are evaluated by calculating eigenvalues and eigenvectors of the variance-covariance matrix related to the point i and its neighbors. The variance-covariance matrix \mathbf{M}_i of the point i is shown in equation (7).

$$\mathbf{M}_i = \frac{1}{|i^*|} \sum_{j \in i^*} (\mathbf{p}_j - \bar{\mathbf{p}}_i)(\mathbf{p}_j - \bar{\mathbf{p}}_i)^T \quad (7)$$

where \mathbf{p}_i = position of point i

i^* = a set of the neighbor points of point i

$\bar{\mathbf{p}}_i$ = barycenter of i^*

We denote eigenvalues of the \mathbf{M}_i by λ_1^i , λ_2^i and λ_3^i ($\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i$) and the corresponding unit eigenvectors by \mathbf{e}_1^i , \mathbf{e}_2^i and \mathbf{e}_3^i respectively. The local distribution of neighbors of point i is figured out by the magnitude relation of the eigenvalues.

When point i is on the pole-like object, the maximum eigenvalue λ_1^i is very large compared with other eigenvalues λ_2^i , λ_3^i , and the eigenvector \mathbf{e}_1^i represents the axial direction of the pole-like objects. On the other hand, when point i is on the planar object, λ_1^i and λ_2^i become relatively large compared with λ_3^i . When point i is on the other object, there are not so many differences between the three eigenvalues.

In order to investigate the magnitude relation of eigenvalues, we compute the three distribution features of each point i using equation (8).

$$\begin{aligned} S_1^i &= \lambda_1^i - \alpha \lambda_2^i \\ S_2^i &= \lambda_2^i - \lambda_3^i \\ S_3^i &= \beta \lambda_3^i \end{aligned} \quad (8)$$

where α, β = adjustment coefficient

When point i is on the pole-like object or planar object or other object, the distribution futures S_1^i , S_2^i and S_3^i becomes the maximum respectively. Fig.5 shows the result of the point classification. Fig.5(a)-(c) shows the utility pole, street sign, and tree respectively. The top row represents the input point clouds. The bottom row represents the point classification result after smoothing.

Vandapel, et al. (Vandapel 2004) classify points by using PCA similarly. They mentioned that it is difficult to hand-tune thresholding classification because those eigenvalues may vary considerably depending on the type of terrain, the type of sensor, and the configuration of the sensor and the vehicle. For this reason, they use machine learning to classify points. However, through experiments using some MMS data, we observed that the points are well classified after smoothing by comparing S_1^i , S_2^i and S_3^i in equation (8), which adopts weights α and β in the original definition of the saliencies (Vandapel 2004). Therefore, we use the direct comparisons of the S_1^i , S_2^i and S_3^i in equation (8) to classify the points (α and β are set to 10, 100 respectively according our experiments).

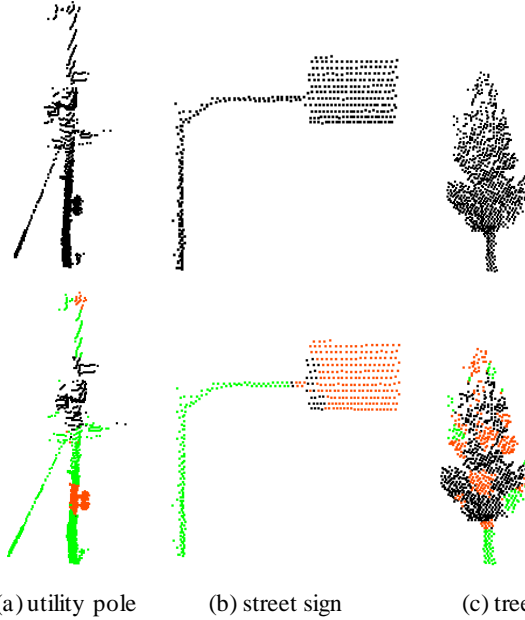


Figure 5. Result of the point classification (green: points on the pole-like object, orange: points on the planar object, black: points on others)

3.5 Pole-like Object Recognition

Finally by evaluating the geometric properties and classified points of the segments, each segment is classified into pole-like object and others. As the minimum requirement for the pole-like object, we assume that the height of the segment is more than 2m which is higher than average human height, and that the number of points of the segment is over 50. Hence, the segments which do not satisfy these conditions are recognized as other objects. In addition, the segment in which over 70% of points are the ones on the other objects is recognized as other objects, because the pole-like objects we intended consist of poles and plane surfaces. For each remaining segment, the degree of the pole-like objects is evaluated by equation (9).

$$f_n = \left(w_1 \frac{|C_n|}{|S_n|} + w_2 \frac{|D_n|}{|C_n|} \right) \times \frac{100}{w_1 + w_2} \quad (9)$$

where w_1, w_2 = weights
 S_n = a set of points of segment n
 C_n = a set of points on the pole-like objects, and included in S_n
 D_n = a set of points that have almost vertical \mathbf{e}_1^i , and included in C_n

An example of the sets of points S_n, C_n and D_n for a street sign is shown in Fig.6. The first term of equation (9) represents the ratio of the points on the pole-like objects in the segment. The second term of equation (9) represents the ratio of the points whose neighbors are distributed vertically in the points on the pole-like objects. The degree of the pole-like object for each segment is evaluated by the weighted sum of the two terms. Fig.7 shows the histogram of the value f_n about various objects in MLS point clouds. The value f_n becomes larger for the almost segments of pole-like objects (In the experiment, it was observed that the segments of the pole-like object had f_n over about 45). Finally, the segments which have f_n larger than the threshold are recognized as pole-like objects.

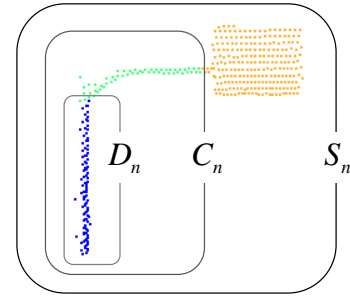


Figure 6. Example of the point clouds of a street sign (orange: points on the planar object, green: points on the pole-like object, blue: points whose neighbors are distributed vertically)

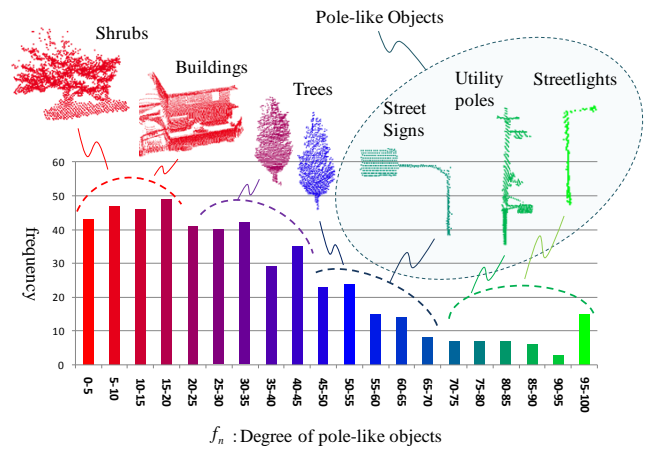


Figure 7. Histogram of f_n , and objects in MLS point clouds

4. RESULT

4.1 Data and Parameter Settings

Three kinds of urban point clouds acquired by MLS system shown in Table 1 were used in our experiments (Ishikawa 2009). As described in section 3.1, the ground points were removed manually from the given data.

In our methods, each parameter is set as follows.

Number of neighbors: $k = 15$

Search radius: $r = 0.5\text{m}$

Number of iterations in smoothing: $n = 40$

Threshold of degree of pole-like object: $\tau = 45$

Weights in equation (9): $w_1=1.0$, $w_2=2.0$

These parameters were determined based on the experiments for some data set.

Data	Measurement Place	Number of Point
I	Kyoto, Japan	939,491
II	Kyoto, Japan	743,357
III	Tokyo, Japan	979,881

Table 1. Data used in the article (without the ground points)

4.2 Effectiveness of the Smoothing

Fig.8 shows the results of point classification. Fig.8(a)-(c) represents a streetlight, street sign, and utility pole respectively. The top row represents the classification result without smoothing. The bottom row represents the classification result with smoothing. The results show that smoothing of the k-nearest neighbors graph improves the classification accuracy of the points by PCA and robustness for the differences of the radius of the object.

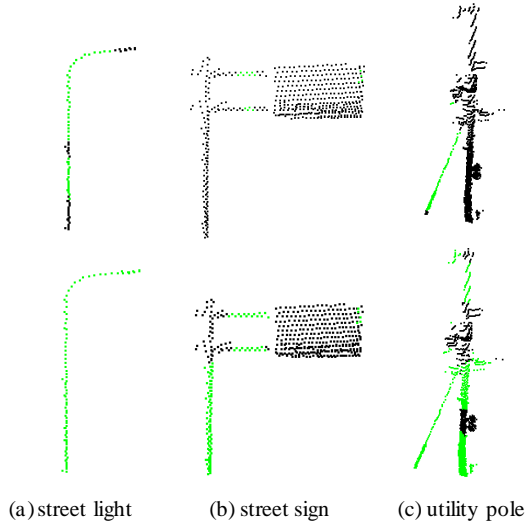
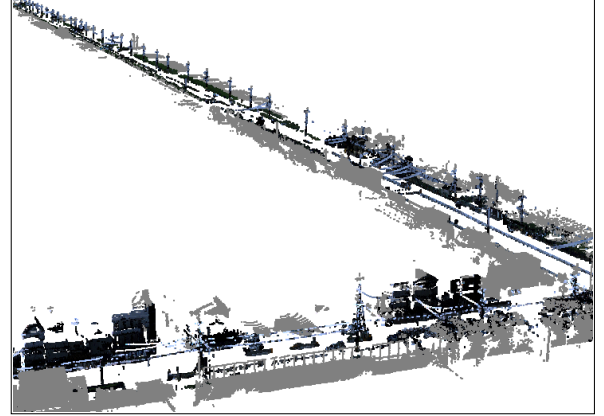


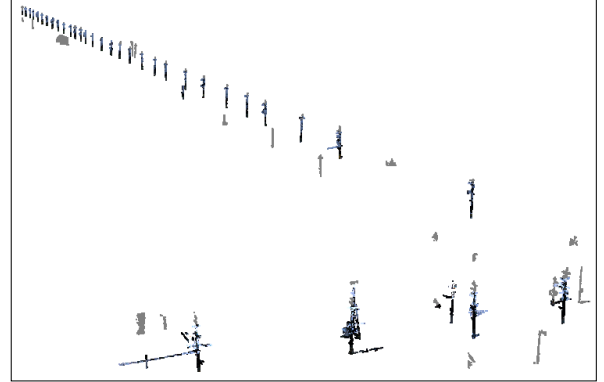
Figure 8. Effectiveness of the smoothing (green: points on the pole-like objects, black: points on the planar objects and others)

4.3 Pole-like Object Recognition Result

Fig.9 shows the pole-like objects recognized by our method. In Fig.9(a), there are a lot of various objects such as trees, buildings, cars, and pole-like objects. In Fig.9(b), the pole-like objects such as streetlights and utility poles are recognized. However, some other objects such as trees are included. The recognition accuracy of the pole-like objects of the three data is shown in Table 2.



(a) input data I



(b) output in our method

Figure 9. Result of the pole-like object recognition

Data	I	II	III
[A] The number of total segments of more than 2m in height, and of $ S_n > 50$	246	121	512
[B] The number of correct pole-like objects	102	51	82
[C] The number of pole-like objects correctly recognized in [B]	80	31	43
[D] Recognition accuracy ($= [C]/[B]$)	78.4 %	60.8 %	52.4 %
[E] The number of correctly segmented objects	68	26	32
[F] The number of pole-like objects correctly recognized in [E]	68	24	32
[G] Recognition accuracy within correctly segmented ($= [F]/[E]$)	100 %	92.3 %	100 %
[H] The number of false recognition	11	7	83

Table 2. Accuracy of the proposed method

The number of correct pole-like objects [B] was manually verified by using the photograph. [H] is the number of objects which were recognized as pole-like objects though they are not the pole-like objects.

The average accuracy of the pole-like object recognition is 63.9% for all segments. On the other hand, the average accuracy of pole-like object recognition is 97.4% for correctly created segments. The accuracy varies depending on the complexity of measured area. The Data I is the point clouds of simple area, but the Data III is the point clouds of complex area which includes overcrowded various objects. Some examples of the objects which were not correctly recognized in the Data III are shown in Fig.10. In both cases, the segment includes pole-like objects and their neighboring objects, such as a hedge and tree. Our recognition method is designed for correctly segmented point clouds, therefore recognition failed for the incorrect segments. The [G] in Table.2 shows our method works well for the correctly created segments. In the future, we will apply an appropriate segmentation method such as that proposed by Golovinskiy et al. (Golovinskiy 2009a) to the inputs and evaluate the recognition rate of the object.

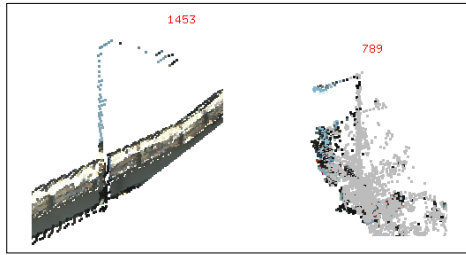


Figure 10. Undetectable pole-like objects by our method

Our method is implemented using standard PC (OS: Windows7 64bit, CPU: Intel Core i7 3.0 GHz, RAM: 6GB). Processing times of the object recognition are shown in Table 3.

Processing	I	II	III
Creation of kd tree	12	5	13
Segmentation	70	43	88
Smoothing	40	34	40
PCA and Point Classification	40	31	29
Pole-like object recognition	0.01	0.01	0.01
Total running time	162	113	170

Table 3. Running time of our method (sec)

5. CONCLUSION

In this article, we developed an algorithm that automatically recognizes pole-like objects with tilt angles and various radii from MLS point clouds in urban environments. Our method is based on the smoothing and principal component analysis for point clouds, and the evaluation of the degree of pole-like objects for the segments. The accuracy of pole-like objects was on average 97.4% for correctly segmented objects.

Future works are to improve the recognition rate by adopting or developing an appropriate segmentation method, and to classify pole-like objects into more detailed object classes, such as utility poles, streetlights, street signs and so on for supporting facility management from MLS point clouds.

ACKNOWLEDGMENTS

We would like to thank Ishikawa Kiichiro of Waseda University for providing the MLS point clouds.

REFERENCES

- Au, O., Tai, C., Chu, H., Cohen-Or, D. and Lee, T., 2008. Skeleton Extraction by Mesh Contraction. Proceeding of ACM SIGGRAPH, Vol.27, issue 3.
- Golovinskiy, A. and Funkhouser, T., 2009a. Min-Cut Based Segmentation of Point Clouds. IEEE Workshop on Search in 3D and Vision (ICCV).
- Golovinskiy, A., Kim, V. and Funkhouser, T., 2009b. Shape-based Recognition of 3D Point Clouds in Urban Environments. International Conference on Computer Vision, pp2154-2146.
- Ishikawa, K., Amano, Y., Hashizume, T., Takiguchi, J., Shimizu, S., 2009. City Space 3D Modeling Using a Mobile Mapping System (in Japanese). The Society of Instrument and Control Engineers Trans. on Industrial Application, Vol.8, No.17, pp.133-139.
- Lai, K. and Fox, D., 2009. 3D Laser Scan Classification Using Web Data and Domain Adaptation. Robotics: Science and Systems.
- Lehtomaki, M., Jaakkola, A., Hyypä, J., Kukko, A. and Kaartinen, H., 2010. Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. Remote Sensing, Vol.2, pp.641-664.
- Manandhar, D. and Shibasaki, R., 2001. Feature Extraction from Range Data. Proceeding of the 22nd Asian Conference on Remote Sensing, Vol.2, pp.1113-1118.
- Moosmann, F., Pink, O. and Stiller, C., 2009. Segmentation of 3D Lidar Data in Non-flat Urban Environments using a Local Convexity Criterion. Proc. IEEE Intelligent Vehicles Symposium, pp.215-220.
- Vandapel, N., F.Huber, D., Kapuria, A., Hebert, M., 2004. Natural Terrain Classification Using 3-D Ladar Data. Proceeding of International Conference on Robotics and Automation.
- Wang, L. and Chu, H., 2008. Graph Theoretic Segmentation of Airborne LiDAR Data. Proc. SPIE Defense and Security Symposium, pp.69790N-1-10.