Data Augmentation of Classifiers for Components in Industrial Plants Using CAD Models

Kohei Shigeta[1], Daiki Hanai[2] and Hiroshi Masuda[2]

[1]The University of Electro-communications, s2032054@edu.cc.uec.ac.jp
[2]The University of Electro-communications, h.masuda@uec.ac.jp
[1]The University of Electro-communications, Daiki.Hanai@uec.ac.jp

Corresponding author: Hiroshi Masuda, h.masuda@uec.ac.jp

Abstract. Old facilities that require renovation often do not have reliable 3D models or 2D drawings due to repeated refurbishment. In such cases, point clouds captured using a terrestrial laser scanner are useful. To create a 3D model of each component from a point cloud, it is necessary to extract each component from the point cloud and identify the component type. Convolutional neural network (CNN) is effective for identifying component types. However, it is not easy to obtain training data suitable for industrial plants. In this paper, we discuss methods for enhancing training data using 3D CAD models. In our method, depth images and intensity images are generated from CAD models to augment training data. The intensity value is calculated by estimating the function between the irradiation angle and the intensity using actual point clouds. In addition, we augment depth images by randomly removing circular or polygonal regions to simulate occlusions. In our evaluation using intensity and depth images generated from point clouds, our method could successfully improve classification accuracy especially when a sufficient number training data could not be obtained.

Keywords: Point Cloud, Convolutional Neural Network, Terrestrial Laser Scanner, Poiny Processing, Reverse Engineering, Shape Reconstruction.
DOI: https://doi.org/10.14733/cadaps.2022.913-923

1 INTRODUCTION

3D models of industrial plants are very useful for simulating maintenance work, such as layout planning and equipment replacement. However, old facilities that require renovation often do not have reliable 3D models or 2D drawings due to repeated refurbishment. In such cases, it is necessary to measure current plants and create as-is 3D models. In recent years, the terrestrial laser scanner (TLS) has advanced rapidly, and it has enabled to capture high-density point clouds of industrial plants. The 3D shape of an industrial plant is captured as hundreds of millions or billions of points.

So far, many methods have been proposed for creating the 3D model of an object from a point cloud [1],[3]. However, point clouds of an industrial plant contain a lot of components with various

sizes and shapes. Each component needs to be identified and extracted from point clouds before creating a 3D model. In addition, missing points cannot be avoided for the point cloud of each component, because many components are often densely placed in industrial plants and the scanning positions are limited. Although small missing points can be repaired by interpolation [8], restoration of large missing portions requires knowledge of target objects.

To create the 3D model of each component from partial point clouds, several methods have been proposed [2], [13],[20]. Jian et al. [4] used a pre-built library of CAD models to created 3D models of components in industrial plants. Joohyuk et al. [8]created skeletons of pipelines from point clouds. Other methods used the industrial standard of components. Many standard components in industrial plants consist of primitive surfaces, such as planes, cylinders, cones, spheres, and tori. Therefore, primitive surfaces are often extracted from point clouds to detect each component. For industrial plants, surface types are typically limited to planes, cylinders, and spheres that can be robustly extracted from partial point clouds [10-11]. Components consisting of other surface types are estimated using connected components[7],[12]. The shapes of the elbow, tee, reducer can be uniquely determined according to the standards if the diameters of connected pipes are given. However, if the component is not standardized, such as valves and manometers, the component type cannot be identified using connected components. In addition, as shown in Figure 1, valves and flanges have various shape patterns when they are assembled. It is difficult to identify specific patterns only using the connection relationship. When component types cannot be estimated from detected surfaces, it is effective to use machine learning to identify them.

In recent years, object recognition using deep learning has been used in various fields and has achieved successful results. There are two approaches for object recognition using point clouds. One method is 2D object recognition using 2D images generated from point clouds, and the other is 3D object recognition using 3D coordinates of point clouds. In 2D object recognition, 2D images are generated by projecting points on 2D images from multiple directions[21]. However, in the case of industrial plants, it is difficult to create images from many directions due to missing points on the back side of each object. In 3D recognition, 3D voxels are often used for convolutional neural network (CNN)[24]. However, it is difficult to determine the appropriate voxel size, because there are various sizes of components in industrial plants. In recent years, CNN models that directly use 3D coordinates have been intensively studied. PointNet [16] and PointNet++ [17] are popular as such methods, and many derivatives have also been proposed[14-15], [17], [22]. However, PointNet was not very powerful in recognizing partial point clouds, such as components in industrial plants [18]. In most existing studies, PointNet has achieved excellent results for complete point clouds with no missing portions. Unfortunately, this condition cannot be satisfied for industrial plants.

In our previous work [18], we identified components in industrial plants by using depth images, intensity images, and RGB images. Since point clouds captured by terrestrial laser scanning are structured as described in a later section, each point cloud can be mapped on the 2D image. Two-dimensional object recognition is mature, and CNN in particular has achieved excellent scores for image recognition. Therefore, for point clouds acquired using TLS, mature pre-trained models, such as VGG16 [19] and ResNet[6], can be used to identify components in industrial plants. However, CNN requires a large amount of labeled data for training classifiers. Obtaining point clouds of many industrial plants is actually difficult, and creating training data can be very costly and time consuming. In addition, it is difficult to prepare training data for plant-specific components. To solve this problem, it is desirable to train the CNN using CAD models rather than labeled point clouds captured from actual industrial plants.

In this paper, we discuss data augmentation using CAD models. CAD models are suitable for generating depth images by projecting points on a 2D plane [18], and such images can be used to augment training data for the CNN classifier based on depth images. However, depth images generated from CAD models are too clean compared to actual point clouds. In this paper, we investigate methods for improving recognition accuracy by modifying depth images generated from CAD models. Then, we also discuss the method for generating intensity images from CAD models to augment the training data for CNN classifier based on intensity images. Since the definition of the

intensity value differs depending on the scanner type, we estimate intensity values by analyzing actual point clouds of industrial plants. In our experiments, we evaluate four types of classifiers using intensity images and depth images and compare their accuracy.

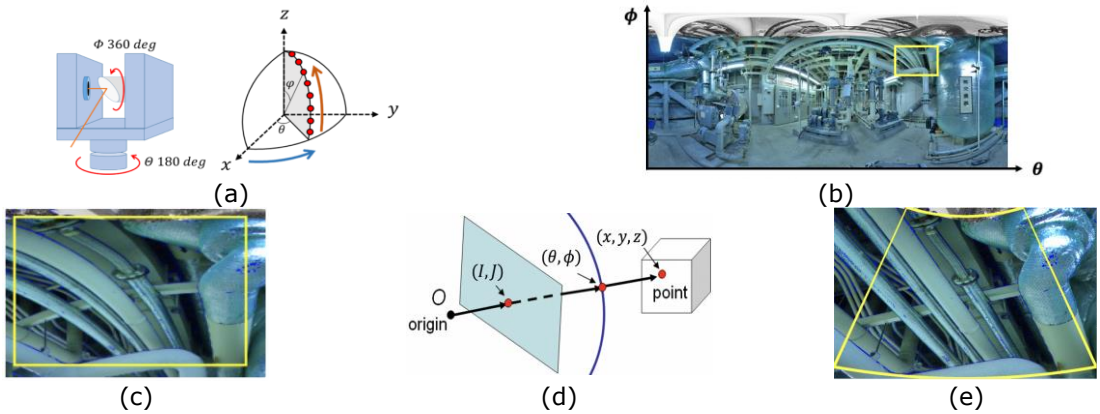
2 GENERATIONS OF 2D IMAGES FROM POINT CLOUDS

Coordinates acquired by TLS are calculated from the travelling distance, the azimuth angle θ, and the zenith angle ϕ of the laser beam, as shown in Figure 2(a). Therefore, each point can be mapped on the (θ,ϕ) plane. Figure 2(b) shows the image of a point cloud. However, the image defined by θ and ϕ is distorted, as shown in Figure2(c).

Due to this distortion, objects in the image are different from actual shapes. Therefore, we correct the distortion by perspective projection, as shown in Figure 2(d). The origin is the source of the laser beam. In this transformation, the coordinate, intensity, and RGB of pixel (I,J) on the perspective image is copied from pixel (θ,ϕ). The transformed image is shown in Figure 2(e).
Since each pixel has the 3D coordinate, intensity, and RGB, the depth images, the intensity images, and the RGB images can be generated from a point cloud, as shown in Figure 3.

(a) (b)

Figure 1: Shape variations: (a) Flange and (b) Valve.



(a) (b) (c) (d) (e)

Figure 2: Generation of 2D images from point clouds: a) Angles of laser beam, (b) RGB image on the (θ,ϕ) plane. (c) Close-up, (d) Perspective projection, and (e) Perspective image.

(a) (b) (c)

Figure 3: 2D Images created from point clouds: (a) RGB image, (b) Intensity image, and (c) Depth image.

3 GENERATION OF DEPTH IMAGES AND INTENSITY IMAGES FROM CAD MODEL

3.1 Generation of Dense Points on CAD Models

Since it is difficult to collect a sufficient number of point clouds of industrial plants, the number of training data is limited. In particular, the number of some components, such as tees and manometers, is very small. When a sufficient number of training data cannot be obtained, overfittings to specific data will occur.

Therefore, we augment training data by creating additional point clouds using CAD models. We created CAD models of the standard components, such as elbows and tees, according to the industrial standards. We also created non-standard components, such as valves and manometers, by measuring their dimensions from point clouds. When there are several assembly shapes such as flanges, we measured the point cloud and created a CAD model.

Figure 4 shows the process of image generation from a CAD model. First, dense points are generated on the surfaces of the CAD model. Then, a triangle face is randomly selected with a probability proportional to the area. Suppose the areas of triangle faces are $\{s_i\}$ $(i = 1, .., n)$, and Σ_m is defined as the sum of $s_1, .., s_m$.Then, a uniform random floating number x is generated in the range of [0,1]. Triangle m is selected only if $\Sigma_m/\Sigma_n \leq x < \Sigma_{m+1}/\Sigma_n$. Then, using the vertices A, B, and C of the triangle and the two random values α and β in $[0, 1]$, the point P is generated on the triangle as follows:

$$P = \left(1 - \sqrt{\alpha}\right)A + \sqrt{\alpha}(1 - \beta)B + \sqrt{\alpha}\beta C$$

Figure 4(b) shows points generated on a CAD model. The generated points are rotated randomly around the X, Y, and Z axes and translated randomly. Depth images are generated by projecting the points on the perspective plane and selecting points that can be seen from the scanner position, as shown in Figure 4(c). Figure 4(d) shows the depth image, in which the depth value is maintained for each pixel. By repeating random rotations and translations, many depth images can be generated from a single CAD model.

3.2 Estimation of Intensity of CAD Models

Point clouds have intensity values as well as 3D coordinates. The intensity values of objects are also effective for object recognition[18]. In this section, we discuss the method for generating the intensity image of each component using a CAD model.

In actual point clouds, intensity values are calculated from the strength of the laser beam reflected from the object. Since white objects have high reflectance of the laser beam, their intensity values tend to become large. The intensity value depends on the distance from the scanner, the irradiation angle, the material and color of the object, and so on. The farther the object is from the scanner, the weaker the reflected laser beam. The intensity values are usually calculated using a software tool provided by a scanner vendor to compensate for the effects of distance. As a result, the intensity image looks like a black and white image, as shown in Figure 4(b)

Unfortunately, the algorithm for calculating intensity values is vendor-specific and is not published. Therefore, it is difficult to calculate the intensity images of components from CAD models

Figure 4: Images generated from CAD model: (a) CAD model, (b) Dense points on CAD model, (c) Points projected on the image plane, and (d) Depth images.

even if the materials and colors of components are known. Therefore, we estimate the intensity values using actual point clouds of industrial plants.

We suppose that each component consists of a similar material and has a similar reflectance. The intensity value is decided depending on the distance from the scanner and the irradiation angle of the laser beam. Since the difference in distance from the laser scanner is small within one component, the points of each component are considered to be approximately the same distance. In addition, the intensity values of each component have to be normalized to [0, 1] in order to apply the CNN. Therefore, in this study, we suppose that the intensity values of each component depends only on the irradiation angle.

We estimate the relationship between the irradiation angle and the intensity value using actual point clouds. The irradiation angle is defined as the angle between the laser beam and the surface normal. The normal vector at each point can be calculated by fitting a plane to neighbor points. However, the calculated normal vectors are noisy and unstable. Thus, we extract cylindrical pipes from point clouds, and obtain their normal vectors and intensity values. Cylinders are extracted using the RANSAC method [10] and their equations are calculated using the least squares method [9]. The normal vector at each point on the CAD model is calculated as the normal vector at the nearest point on the cylinder. Then, the irradiation angle is calculated as the angle between the normal vector and the laser beam.

There are a lot of cylindrical pipes in engineering plants and each cylindrical surface contains a wide range of normal vectors. Therefore, many pairs of irradiation angle θ and intensity value I can be obtained from point clouds. In our method, a piecewise polynomial function is used to represent the relationship between the irradiation angle and the intensity value. Suppose that $\{(\theta_i,\ I_i)\}\ \ (i = 1,..,n)$ are obtained from cylinders extracted from point clouds. Then, a piecewise polynomial function f is fitted to $\{(\cos\theta_i,\ I_i)\}$ so that the intensity value is calculated as $I = f(\cos\theta)$. In this study, the piecewise polynomial function is calculated simply by fitting a cubic B-spline curve [23] to $\{(\cos\theta_i,\ I_i)\}$. Intensity images are created by substituting the irradiation angles calculated from the CAD model into the function f. Figure 5 shows intensity images generated from CAD models.

3.3 Augmentation of Point Clouds Generated from CAD Model

Actual point clouds are noisy and may be partially missing due to occlusions, but the depth image generated from a CAD model is unrealistically clean with no missing points. The image-based CNNs typically augment training data by modifying images using rotation, flipping, scaling, and adding noise. However, occlusion is not supported by common tools for the image-based CNNs. Thus, we focus on augmentation using missing points in depth images generated from CAD models. In this study, missing portions are defined as circle or polygonal regions.

For removing a circular region from a depth image, k-nearest neighbor points are removed for a randomly selected point. This process is repeated n times to eliminate n circular regions. Suppose that N is the number of points in a point cloud. With each augmentation, k and n are randomly selected as uniform random numbers in the ranges of $[N \cdot k_{min}, N \cdot k_{max}]$ and $[n_{min,} n_{max}]$, respectively. k_{min} and k_{max} are the ratios of the number of points. In this study, the ranges $[k_{min}, k_{max}]$ and $[n_{min,} n_{max}]$ are specified by the user.

For removing polygonal regions from a depth image, a pixel position is randomly selected on the depth image. Then, m points are randomly selected from a range within the distance d from the selected pixel. A polygon is generated by connecting the selected m points, and the polygonal region is removed from the depth image. This process is repeated n times to eliminate n polygonal regions. With each augmentation, the numbers n and m are randomly selected in the ranges of $[n_{min,} n_{max}]$ and $[m_{min,} m_{max}]$, respectively. In this study, distance d and the ranges for n and m are specified by the user.

Figure 6 shows examples of augmented depth images. In the upper case, circular regions are removed from the depth image. In the bottom case, polygonal regions are removed.

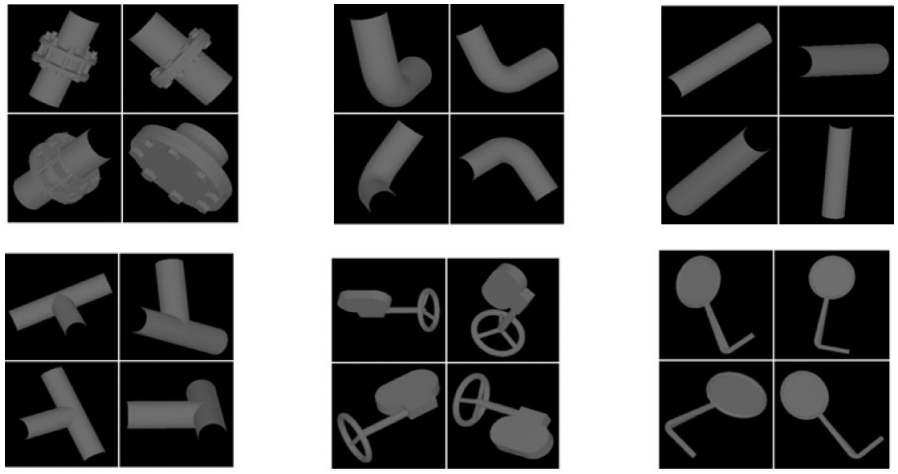


Figure 5: Intensity images created from CAD models.

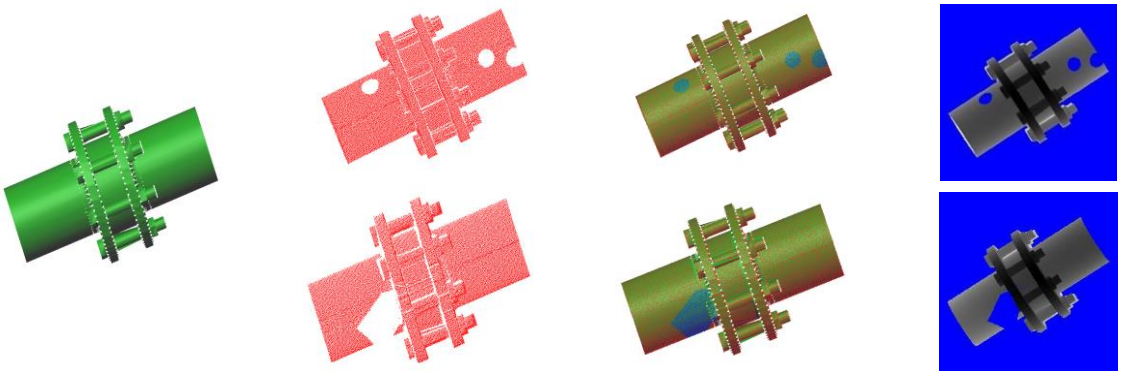


Figure 6: Augmenting virtual point clouds (Upper: Missing using k-nearest neighbors, Bottom: Missing using polygons): (a) CAD model, (b) Partly missing point cloud, (c) Missing portions shown in blue, and (d) Incomplete depth image.

4 EXPERIMENTAL RESULTS

4.1 Evaluation of Intensity Images

In order to evaluate our method, we trained the CNN classifiers using actual point clouds and virtual point clouds generated from CAD models, and verified the accuracy of classification. It is considered that point clouds generated from CAD models are effective especially when the number of training data from actual point clouds is small. Therefore, we evaluated the classifier using different numbers of training data generated from actual point clouds. Table 1 shows the numbers of training data in four cases. The number of virtual training data generated from CAD models is the same in all cases, but the number of training data generated from actual point clouds is different in each case. Table 2 shows the numbers of test data, which were generated from actual point clouds.

When the CNN classifier was applied, the training data were augmented by rotating, inverting, enlarging, changing the brightness value, and adding noise by using a common tool in Keras [25]. We used VGG16 trained with ImageNet [5] as a classifier. This classifier was further trained by

transfer learning using training data generated from point clouds. Then, the dropout layer was added to the output layer side. In this study, the F value was used as an evaluation index.

To evaluate CNN classifiers trained using intensity images, we created two types of classifiers in four cases. One classifier is trained using only the intensity images generated from the actual point clouds. The other classifier is trained using both the actual and virtual intensity images. The results are shown in Table 3. When the number of training data generated from actual point clouds is less than 10, the symbol '*' is added to the component name. In all cases, intensity images generated CAD models were effective to improve the F-measures. In particular, the F-measures has improved considerably for components with a very small number of training data, such as Tees.

	Flange	Elbow	Straight	Tee	Valve	Manometer*	Total
Actual Point Clouds	80	86	40	11	22	4	243
Virtual Point Clouds	120	80	80	80	80	80	520

(a) Number of training data in Case 1

	Flange	Elbow	Straight	Tee*	Valve	Manometer*	Total
Actual Point Clouds	40	43	20	6	11	2	122
Virtual Point Clouds	120	80	80	80	80	80	520

(b) Number of training data in Case 2

	Flange	Elbow	Straight	Tee*	Valve*	Manometer*	Total
Actual Point Clouds	20	22	10	3	6	1	62
Virtual Point Clouds	120	80	80	80	80	80	520

(c) Number of training data in Case 3

	Flange	Elbow	Straight*	Tee*	Valve*	Manometer*	Total
Actual Point Clouds	10	11	5	1	3	1	31
Virtual Point Clouds	120	80	80	80	80	80	520

(d) Number of training data in Case 4

Table 1: Numbers of training data.

Flange	Elbow	Straight	Tee	Valve	Manometer	Total
80	85	40	10	20	3	238

Table 2: Numbers of test data.

	Flange	Elbow	Straight	Tee	Valve	Manometer*	Average
Actual Intensity	91.7%	90.1%	95.1%	73.7%	81.0%	80.0%	89.9%
Actual & Virtual	91.3%	91.7%	97.5%	85.7%	82.9%	80.0%	91.6%

(a) Case 1 (243 training data from actual point clouds)

	Flange	Elbow	Straight	Tee*	Valve	Manometer*	Average
Actual Intensity	89.3%	89.3%	91.6%	42.9%	77.8%	85.7%	86.7%
Actual & Virtual	90.1%	90.6%	94.7%	58.8%	81.1%	80.0%	88.9%

(b) Case 2 (122 training data from actual point clouds)

	Flange	Elbow	Straight	Tee*	Valve*	Manometer*	Average
Actual Intensity	86.2%	84.1%	89.4%	26.7%	75.0%	80.0%	82.5%
Actual & Virtual	87.5%	86.6%	94.7%	70.0%	78.0%	66.7%	86.6%

(c) Case 3 (62 training data from actual point clouds)

	Flange	Elbow	Straight*	Tee*	Valve*	Manometer*	Average
Actual Intensity	81.9%	80.7%	84.6%	0.0%	59.5%	100%	76.8%
Actual & Virtual	83.0%	81.1%	78.8%	62.5%	72.2%	57.1%	79.5%

(d) Case 4 (31 training data from actual point clouds)

Table 3: F-measures of classifiers using intensity images.

4.2 Evaluation of Augmented Depth Images

To evaluate depth images generated from CAD models, we created four types of classifiers. The first classifier was trained using only actual point clouds. The second classifier was trained using both actual and virtual point clouds without augmentation by missing regions. For the third classifier, virtual data were augmented by removing circle regions in the training phase. For the fourth classifier, polygonal regions were randomly eliminated from virtual depth images. In this evaluation, we used training data of Case 1, as shown in Table 1(a).

Table 4 shows F-measures of four classifiers. The results show that virtual depth images were effective to improve F-measures. Data augmentation simulating occlusions was also effective to further improve F-measures. In particular, F-measures were improved considerably when the number of training data was small. In Case 1, F-measures were almost the same when circle regions were removed and when polygonal regions were removed.

Then, we evaluated F-measures when the number of training data was reduced. Table 5 shows F-measures for Case 2 and Case 3 shown in Table 1(b), (c). The result shows that classifiers trained using augmented depth images achieved good F-measures even when the number of depth images generated from actual point clouds was small. In these evaluations, F-measures were better when circle regions were removed.

	Only Actual	Actual & Virtual	Actual & Virtual (Circle Regions)	Actual & Virtual (Polygonal Regions)
Flange	94.3%	97.4%	98.1%	98.1%
Elbow	92.1%	95.5%	96.0%	95.3%
Straight	92.5%	97.5%	96.3%	97.5%
Tee	62.5%	75.0%	84.2%	85.7%
Valve	97.4%	95.2%	100.0%	100.0%
Manometer	100.0%	100.0%	100.0%	100.0%
Average	92.2%	95.6%	96.6%	96.7%

Table 4: F-measures of classifiers using depth images (for Case 1).

	Case 2		Case 3	
	Actual & Virtual (Circle Regions)	Actual & Virtual (Polygonal Regions)	Actual & Virtual (Circle Regions)	Actual & Virtual (Polygonal Regions)
Flange	96.2%	95.6%	96.9%	94.0%
Elbow	95.4%	90.8%	95.2%	90.1%
Straight	96.1%	85.7%	96.3%	96.2%
Tee	85.7%	77.8%	75.0%	60.9%
Valve	90.0%	94.7%	86.5%	90.0%
Manometer	100.0%	100.0%	85.7%	100.0%
Average	95.0%	91.5%	94.3%	91.3%

Table 5: F-measures of classifiers using depth images (for Case 2 and 3).

5 CONCLUSION

A large amount of labeled data are required for training CNN classifiers, but it is difficult to obtain a sufficient number of point clouds of industrial plants. In this paper, we proposed methods for augmenting training data using CAD models. In our method, dense points were generated on a CAD model and they made depth and intensity images. The intensity value was calculated using the relationship between intensity values and irradiation angles, which were estimated using actual point clouds. In addition, depth images were augmented by randomly eliminating circular or polygonal regions to simulate occlusions in actual point clouds. In our experimental results, we evaluated classifiers based on intensity images and depth images. In both evaluations, the classification accuracy was improved using virtual data especially when the number of actual data was very small.

There are several remaining issues for our method. In our experiments, the classifier trained by only virtual data did not work well, and virtual data had to be used with actual data even if the number of actual data was small. We consider that virtual data lacks some essential features of actual data. We would like investigate this issue in more detail. Many CNN-based segmentation methods have been proposed so far, but they are not suitable for robustly extracting incomplete point clouds of components in industrial plants. We would also like to investigate methods for extracting components from large-scale point clouds of industrial plants.

Kohei Shigeta, https://orcid.org/0000-0002-4136-4199
Daiki Hanai, https://orcid.org/0000-0002-7918-9103
Hiroshi Masuda, https://orcid.org/0000-0001-9521-6418

REFERENCES

[1] Alliez, P.; Cohen-Steiner, D.; Tong, Y.; Desburn, M.: Voronoi-based Variational Reconstruction of Unoriented Point Sets, Eurographics Symposium on Geometry Processing, 2007, 39-48. https://dx.doi.org/10.2312/SGP/SGP07/039-048

[2] Bey, A.; Chaine, R.; Marc, R.; Thibault, G.; Akkouche, S.: Reconstruction of Consistant 3D CAD Models from Point Cloud Data Using a Priori CAD Models, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-5/W12, 2011, 289-294. https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-289-2011

[3] Calakli, F.; Taubin, G.: SSD: Smooth Signed Distance Surface Reconstruction, Computer Graphic Forum, 30(7), 2011, 1993-2002. https://doi.org/10.1111/j.1467-8659.2011.02058.x

[4] Chai, J.; Chi, H.; Wang, X.; Wu, C.; Jung, K. H.; Lee, J. M.: Automatic as-built modeling for concurrent progress tracking of plant construction based on laser scanning, Concurrent Engineering, 24(4), 2016. https://doi.org/10.1177/1063293X16670499

[5] Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database, IEEE, 2009. https://ieeexplore.ieee.org/document/5206848

[6] He, K.; Zhang, X.; Ren, S.; Sun, J: Deep Residual Learning for Image Recognition, Computer Vision and Pattern Recognition, 2016. https://doi.org/10.1109/CVPR.2016.90
[7] Kawashima, K.; Kanai, S.; Date, H.: As-built modeling of piping system from terrestrial laser-scanned point clouds using normal-based region growing, Journal of Computational Design and Engineering, 1(1), 2014, 13-26. https://doi.org/10.7315/JCDE.2014.002
[8] Lee, J.; Son, H.; Kim, C.; Kim, C.: Skeleton-Based 3D Reconstruction of As-Built Pipelines from Laser-Scanned Data, Automatic in Construction, 35, 2013, 199-207. https://doi.org/10.1016/j.autcon.2013.05.009
[9] Lukács, G.; Ralph, M.; Dave, M.: "Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation." European Conference on Computer Vision, 1998, 671-686. https://doi.org/10.1007/BFb0055697
[10] Masduda, H.; Niwa, T.; Tanaka, I.; Matsuoka, R.: Reconstruction of polygonal faces from large-scale point-clouds of engineering plants, Computer-Aided Design and Applications, 12(5), 2014, 150-152. https;//dx.doi.org/10.14733/cadconfp.2014.150-152
[11] Masuda, H.; Tanaka, I.; Enomoto, M.: Reliable Surface Extraction from Point-Clouds using Scanner-Dependent Parameters, Computer-Aided Design and Applications, 10(2), August 2013, 265-277. 10.3722/cadaps.2013.265-277
[12] Matsuoka, R.; Masuda, H.: Reconstruction of Structure Shapes of Facilities from Large-scale Point Cloud (1st report) - Estimation and evaluation of connecting components based on extracted surfaces -, Seimitsukougakukaishi, 2014, 604-608. 10.2493/jjspe.80.604
[13] Mizoguchi, T.; Kuma, T.; Kobayashi, Y.; Shirai, K.: Manhattan-World Assumption for As-Built Modeling Industrial Plant, Key Engineering Materials, 523-524, 2012, 350-355. https.//doi.org/10.4028/www.scientific.net/KEM.523-524.350
[14] Qi, C. R.; Chen, X.; Litany, O.; Guibas, L. J.: ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes, Conference on Computer Vision and Pattern Recognition, 2020, http://dx.doi.org/10.1109/CVPR42600.2020.00446
[15] Qi, C. R.; Liu, W.; Wu, C.; Su, H.; Guibas, L. J.: Frustrum PointNets for 3D Object Detection from RGB-D Data, Conference on Computer Vision and Pattern Recognition, 2018, 918-927. https://doi.org/10.1109/CVPR.2018.00102
[16] Qi, C. R.; Su, H.; Mo, K.; Guibas, L. J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Conference on Computer Vision and Pattern Recognition, 2017, 652-660. https://doi.org/10.1109/CVPR.2017.16
[17] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Conference on Neural Information Processing System, 2017. https://dl.acm.org/doi/10.5555/3295222.3295263
[18] Shigeta, K; Masuda, H: Extraction and Recognition of Components from Point Clouds of Industrial Plants, Computer-Aided Design & Applications, Vol. 18, No. 5, 2021, 890-899. 10.14733/cadaps.2021.890-899
[19] Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, Eighth International Conference on Learning Representations, 2015. https://dblp.org/rec/journals/corr/SimonyanZ14a.html
[20] Son, H.; Kim, C.; Kim, C.: Automatic 3D Reconstruction of As-Built Pipeline Based on Curvature Computations from Laser-Scanned Data, Construction Research Congress, 2014, 925-934. https://ascelibrary.org/doi/10.1061/9780784413517.095
[21] Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E.: Multi-view Convolutional Neural Networks for 3D Shape Recognition, International Conference on Computer Vision, 2015, 945-953. https://doi.org/10.1109/ICCV.2015.114
[22] Ye, X.; Li, J.; Hexiao, H.; Du, L.; Zhang, X.: 3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation, Proceedings of the European Conference on Computer Vision, 2018, 403-417. https://doi.org/10.1007/978-3-030-01234-2_25
[23] Yuwen, S.; Dongming, G.; Zhenyuan, J.; Weijun, L.: B-spline surface reconstruction and direct slicing from point clouds, The International Journal of Advanced Manufacturing Technology, 2006, 918-924. https://doi.org/10.1007/s00170-004-2281-6

[24] Zhou, Y.; Tuzel, O.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, Conference on Computer Vision and Pattern Recognition, 2018. https://doi.org/10.1109/CVPR.2018.00472
[25] Francois, C: Keras, 2015. https://keras.io/