www.cadanda.com
Computer-Aided Design and Applications™

Extraction and Recognition of Components from Point Clouds of Industrial Plants

Kohei Shigeta[1] and Hiroshi Masuda[2]

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

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

Abstract. Point clouds of industrial plants are very useful for supporting renovation planning, production and product design, asset management, and so on. However, point clouds of an industrial plant contain a large number of components. In order to utilize point clouds, it is necessary to extract each component from point clouds and identify its type. In this paper, we discuss methods for identifying component types in industrial plants using machine learning. In our method, cylinders and planes are detected from point-clouds and candidate component regions are extracted. Since point clouds captured using the terrestrial laser scanner can be mapped on the 2D grid, convolutional neural network (CNN) designed for images can be applied. Three types of 2D images are generated from point clouds, and they are used for classification. To increase the numbers of training data, depth images are augmented using CAD models. In evaluation, nine classifiers were trained and evaluated. By comparing the nine CNN models, we discuss classifiers suitable for recognizing components in industrial plants.

Keywords: Point-Cloud, Convolutional Neural Network, Terrestrial Laser Scanner, Point Processing, Reverse Engineering, Shape Reconstruction.
DOI: https://doi.org/10.14733/cadaps.2021.890-899

1 INTRODUCTION

In recent years, rapid advances in terrestrial laser scanners have made it easier to acquire dense point-clouds from large industrial plants. Due to repeated renovations of existing industrial plants, reliable 3D models or drawings cannot be obtained in most cases. Point clouds are very useful for obtaining faithful 3D data of current industrial plants, and they can be used for renovation planning, production and product design support, asset management, and so on.

However, point clouds of an industrial plant contain a large number of components. Therefore, in order to utilize point clouds, it is necessary to extract each component from point clouds and identify its type. Many reverse engineering methods have been developed for generating 3D models from point-clouds [6],[20]. Although most existing methods require point-clouds without large missing portions, it is difficult to acquire complete point-clouds of industrial plants unlike measuring mechanical parts. Since many components are densely placed in industrial plants and the terrestrial laser scanners are relatively large, point clouds can be acquired only from limited locations. Therefore, it is necessary to detect each component in industrial plants from incomplete point-clouds. Interpolation methods can repair small missing portions [19], but such methods often produce incorrect shapes from an engineering point of view when interpolating large missing portions.

Various methods have been developed for generating 3D models of industrial components from incomplete point clouds [1],[8],[10],[11],[22]. For extracting components of industrial plants, primitive surfaces are typically extracted from point clouds. Many standard components are commonly used in industrial plants, and they mainly consist of primitive surfaces such as planes, cylinders, cones, spheres and tori. In most practical reverse engineering systems for industrial plants, primitive surfaces are limited to planes, cylinders, and spheres, because they have a small number of degrees of freedom and their parameters can be calculated accurately from partial point clouds. Standard components with torus or conical surfaces are estimated using industry standards for connecting pipes [18]. Figure 1(a) shows standard components that could be estimated using connecting cylindrical pipes [13]. These shapes can be uniquely determined according to the industrial standards if the radii, axes and end locations of connecting pipes are given. For example, the torus surface of an elbow can be uniquely determined by the radii of two connecting pipes according to the industrial standards.

However, this approach cannot handle non-standard components, such as valves and manometers. In addition, when standard components have some shape variations, this approach may fail to identify component types. For example, flanges may be composed of multiple plates connected by shafts, and pipes may be wrapped with thermal insulation. Furthermore, as shown in Figure 1 (b), when there are multiple interpretations of pipe routes, existing methods may produce incorrect components.

When component types cannot be identified only from cylindrical and planar surfaces, machine learning methods are promising, because one of the most common applications of machine learning is object type recognition. In recent years, deep learning has achieved great success in object recognition, and it is becoming a commonly used tool in many fields.

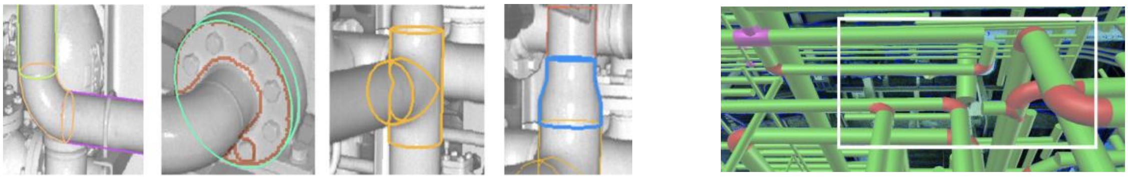
Several methods have been proposed for applying deep learning to point clouds [7], [14], [23]. Point-based methods can be mainly categorized into three approaches. One approach is projecting a point cloud of an object onto 2D planes from various directions, and generating many 2D images of the object. In this approach, well-trained convolutional neural network (CNN) designed for images can be applied. However, in cases of incomplete point clouds, correct 2D images can be obtained only from the scanner position, and therefore, a variety of 2D images cannot be obtained.

In the second approach, 3D voxel structure is generated from point clouds, and CNN is applied to the 3D array of voxels. However, 3D voxel representation cannot maintain detail shapes of components. In large industrial plants, there are various sizes of components, and the point density largely vary according to the distance from the scanner position. Therefore, it is not easy to generate appropriate voxel structure suitable for recognizing a variety of components.

The third approach is projecting a point cloud onto 1-D lines from many directions using 1-D convolution and normalization. In recent years, PointNet [2] has been intensively studied as a deep learning model of unorganized point clouds. This model consists of multiple layers of 1-D convolution, normalization, and max pooling. Derivative methods of the PointNet model have also been proposed such as the method combined with 2D images [5] and semantic segmentation methods [3-[4]. We evaluated PointNet for components in industrial plants, but this model could not achieve good recognition results in our experiments. This is because an insufficient number of training data can be obtained and only partial point clouds can be available for components in industrial plants. Therefore, it is required to develop methods suitable for recognizing components in industrial plants from point clouds captured using the terrestrial laser scanner.

In this paper, we discuss methods for recognizing components in point clouds using deep learning. This research aims to solve the following two problems by using machine learning.

(1) When estimating the types of components from only the detected planes and cylinders, the types may be incorrectly estimated, as shown in Figure 1 (b). In this research, we aim to improve the estimation accuracy using features automatically extracted by deep learning.



(a)

(b)

Figure 1: Component detection using cylinder and plane: (a) Standard components extracted from cylinders and planes, and (b) Ambiguous pipe routing.

(2) Non-standard components, such as valves and thermometers, are difficult to identify using existing methods. This paper investigates whether non-standard components can be identified using machine learning even if planes and cylinders cannot be detected for the component.

Our method is complementary to existing methods. In our method, first, cylinders and planes are detected from point-clouds and candidate component regions are extracted. For recognizing component types, three types of images are generated using RGB, intensity, and coordinates, which are commonly output by terrestrial laser scanners. Since only a small number of training data are available for some component types, training data are augmented using CAD models of components. Then, three types of classifies are integrated for identifying component types. In our experiments, we evaluate nine CNN models including PointNet and discuss methods suitable for recognizing components in industrial plants.

2 COMPONENTS EXTRACTION

2.1 Detection of Planes and Cylinders

The terrestrial laser scanner emits laser beams, whose directions are determined by the azimuth angle θ and the zenith angle ϕ, as shown in Figure 2(a). Since the angle intervals are constant, points can be mapped on the 2D grid defined by θ and ϕ. Therefore, each point-clouds can be converted into a 2D image. Figure 2(b) is an intensity image in which each pixel represents an intensity value of a point.

When a point cloud is captured using the terrestrial laser scanner, each point has an intensity value and an RGB color as well as a coordinate. The intensity value represents the strength of a returned laser beam. The intensity value is larger for white objects and smaller for black objects. The intensity value is represented by an integer or a floating-point number depending on the laser scanner type. In this paper, the intensity value is normalized to a floating-point number between 0 and 1. RGB colors are captured using a camera built in the laser scanner, and added to points as attributes in the post process. In this paper, each RGB color is represented by three floating-point numbers between 0 and 1.

Planes and cylinders can be efficiently detected on the 2D grid using the method proposed by Masuda, et al. [16-[17]. In this method, a point cloud is segmented into connected graphs by connecting adjacent points on the grid only if the distance between adjacent points is smaller than a threshold. The threshold is adaptively determined according to the distance from the laser scanner. In Figure 3(b), connected graphs are shown in different colors. Planes and cylinders are detected from each connected graph using the RANSAC method. Since the performance of the RANSAC method depends on the number of points in each connected graph [21], the connected graph is further subdivided into smaller ones each time a cylinder or a plane is detected and removed. The detection and subdivision are repeated until the number of vertices in a connected graph becomes smaller than a threshold. Figure 3(c) and (d) show detected planar and cylindrical regions, respectively.

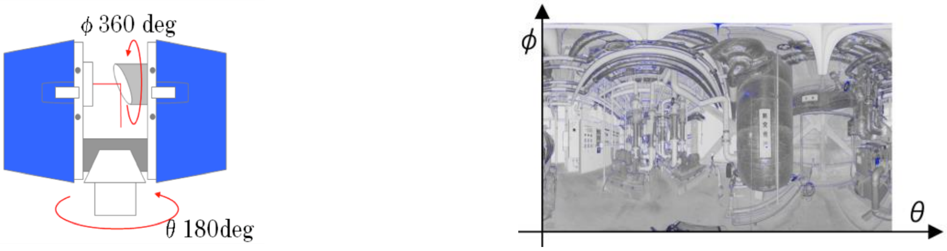
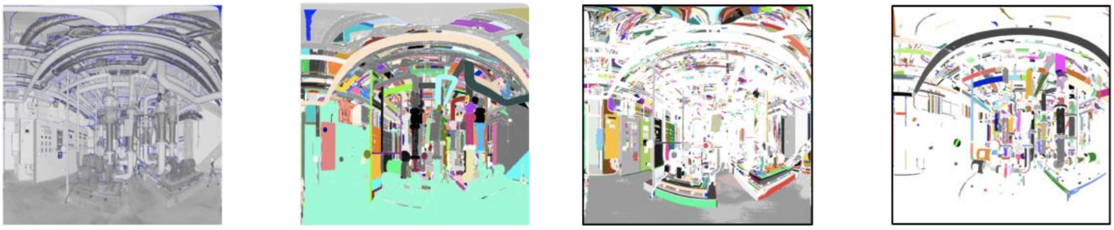


Figure 2: Points arranged on the 2D plane: (a) Scanner rotation angle and (b) Intensity image.



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

Figure 3: Extraction of planes and cylinders: (a) Point cloud, (b) Segmentation, (c) Planar regions, and (d) Cylindrical regions.

Figure 4: Regions connecting to cylindrical pipes.

2.2 Detection of Regions Connecting to Cylindrical Pipes

When component types are estimated using connected cylindrical pipes, variations of detectable component types are limited to the nominal shapes of standard components. In this research, we extract components connecting to cylindrical pipes and identify their types using machine learning.

There are two problems in extracting component regions. One problem is that components far from the laser scanner have low point density and high noise levels. In such cases, cylindrical surfaces may be extracted as planes. The other problem is the cylindrical surface of a pipe may partly include points on connecting components at the boundaries with other components. In such cases, points of connecting components are partly missing and object recognition becomes difficult.

In order to solve these problems, we regard unreliable pipes and regions near pipe boundaries as candidate component regions. Machine learning is applied to candidate component regions. In our method, a point cloud on the 2D grid is segmented into pipe regions, large planar regions, and candidate component regions. Since floors, walls, ceilings, plates, or housing boxes of equipment are composed of large planes, planar regions with a large number of points are removed from candidate component regions. When the length of a cylindrical pipe is too small or the radius is too small or too large, the cylinder may be incorrectly extracted. Such cylindrical regions are removed from pipes and added to candidate component regions. Points near the boundaries of pipes are also added to candidate component regions.

In Figure 4, pipe regions are shown in green, planar regions in blue, and candidate component regions in red. Then, candidate component regions are segmented into connected graphs. The region of each connected graph is enlarged so that pipe regions are partly included.

Then, machine learning is applied to each connected graph for identifying the component type. Dimensions of typical components, such as pipes, flanges, tees, and elbows, are specified by the industry standards. If the 3D shape of a component is determined, equations of planes and cylinders are refined using the industrial standards. In cases of non-standard components, the component type is simply added as an attribute to the point-set represented as a connected graph, because it is difficult to precisely reconstruct the 3D model from partial points.

3 CLASSIFIERS

3.1 Generating Images from Points

Since each connected graph is represented on the 2D grid, CNN for images can be used for classification. However, since 2D images defined using the azimuth angle θ and the zenith angle ϕ are distorted, straight lines are represented as curved lines. Distortion is particularly large at the top and bottom in the 2D grid. As a result, images of the same type of components may be significantly different depending on their locations. This distortion may prevent correct classification of components.

To solve this problem, the angle coordinate (θ, ϕ) is converted to the perspective coordinate (I, J). Perspective projection maps a point on a unit sphere onto the perspective projection plane, as shown in Figure 5(a). The perspective plane is defined to be perpendicular to the line connecting the center of points and the scanner position. By generating images using the perspective projection, the straightness of lines can be preserved, as shown in Figure 5(b). We note that the 2D image is generated only from the scanner position because the points of each component are only partially captured and the adequate silhouette shape can be obtained only from a specific direction.

Since each point has a 3D coordinate, an intensity value, and an RGB color, three types of images can be generated from point clouds using the perspective projection. We call three images as the intensity image, RGB image, and depth image. Figure 6 shows RGB and intensity images. The depth image is generated using the distances from the scanner position. When the distance d is calculated for each point in a connected graph, the depth value is normalized between 0 and 1, as $(d - d_{min})/(d_{max} - d_{min})$. The normalization is required for the convergence of CNN.

3.2 Data Augmentation

For training a CNN classifier, a large number of images are required. In this paper, for obtaining training data, we extract components from point clouds captured from a few industrial plants. To increase the number of training data, we apply data augmentation to three types of images.

For RGB and intensity images, we can use common techniques for image augmentation. In image data augmentation, each image can be modified by rotating, scaling, shrinking, changing brightness values, adding black and white noise, and horizontally flipping. These variations are added to training data. To augment depth images, we create CAD models of typical components and generate depth images from the CAD models. Augmentation using CAD models is effective when the number of components is small, such as tees, valves, and pressure gauges.

For each standard component, a single CAD model is generated. Although each component has various sizes, similar shapes generate the same normalized depth image. For non-standard components, we created CAD models of typical template shapes by observing point clouds.

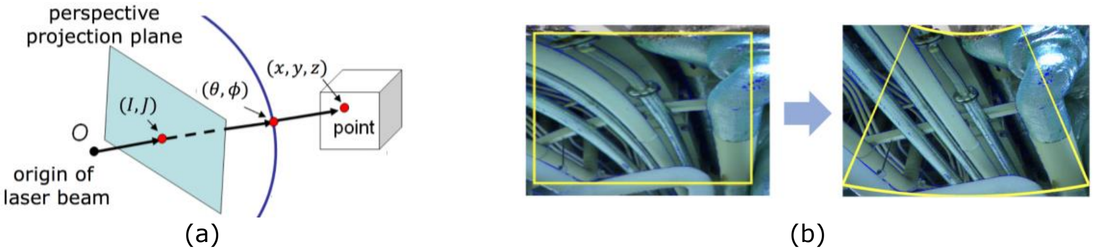


Figure 5: Perspective projection: a) Mapping, and (b) Conversion to the perspective image.

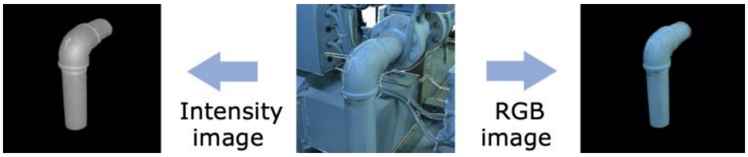


Figure 6: Generation of RGB and Intensity images from point clouds.

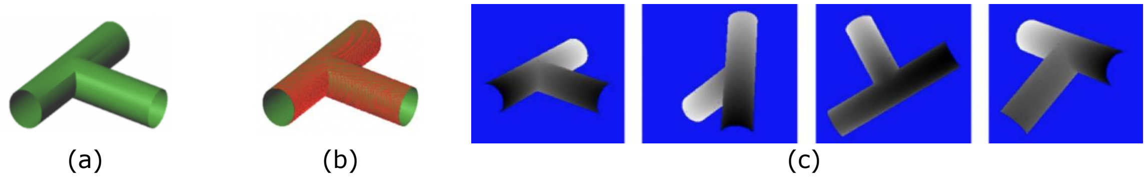


Figure 7: Augmentation by additional depth images: (a) 3D model, (b) Points on faces, and (c) Depth images.

A lot of depth images can be generated from a single CAD model by changing the positions and orientations. In our method, each CAD model is converted into a triangle mesh, and points are generated randomly on each triangle with a probability proportional to the triangle area. Depth images are generated by projecting points onto the perspective projection planes. Figure 7 shows depth images generated from a CAD model.

3.3 Features Extraction and Selection

In our method, RGB images, intensity images, depth images are generated from each connected graph of points. Depth images are also created from CAD models. Three classifiers can be created from each type of images. For comparison, another classifier is created for depth images that are not augmented using CAD models. We call these classifiers as single-input classifiers. We also create the integrated classifier using all types of images. For comparison, we create the integrated classifier using PointNet instead of the classifier based on depth images.

Due to the small number of training data, classifiers are trained using transfer learning. We use VGG16 [12] trained by ImageNet [9] as the base model. The original VGG16 outputs class types from the final layer, and 1024 features from the fully connected layer immediately before the final layer. For single-input classifiers, the three fully connected layers in the VGG16 model are replaced with three unlearned fully connected layers. In transfer learning, the 11th convolutional layer and subsequent layers are updated using training data.

For creating the integrated classifier, 1024 features are extracted from each single-input classifier. Then, effective features are selected using the method proposed by Boruta [15]. Finally, the effective features obtained from three classifiers are concatenated. The class type for each concatenated feature vector can be identified using the random forest method or CNN with fully-connected layers. In this paper, we implement the two methods and compare their results.

4 EXPERIMENTAL RESULTS

To evaluate our method, we extracted points of elbows, flanges, straight pipes, T-shaped pipes, valves, and pressure gauges, as shown in Figure 8. The number of data of each component is shown in Table 1. In point clouds for evaluation, the numbers of manometers, tees, and valves were especially small compared to other components. Then, RGB images, intensity images, and depth images were created from extracted data. We also created CAD models of components. For three types of flange shapes, we created three CAD models. Then, depth images were generated using CAD models. The number of depth images generated from CAD models is shown in Table 1.

For classification, one half of the data were used for training, and the other half were used for evaluation. The accuracy of classifiers was verified using F-measures, which indicate the accuracy as the harmonic mean of precision and recall.

We first evaluated single-input classifiers. For comparison, we also evaluated PointNet [2] using the same point clouds. Table 2 shows the results of single-input classifiers. In this evaluation, the classifier using depth images enhanced with CAD models has achieved the best score. This result shows that the use of CAD models has significantly improved the accuracy of tees, valves and pressure gauges, whose numbers were small. The score of PointNet was the lowest among five classifiers. F-measures were very low especially when sufficient numbers of training data could not be obtained. This might be because data augmentation could not be applied to point clouds and the well-trained learned model was not available for the PointNet model.

In the next experiment, we evaluated the integrated classifiers. Features from three single-input classifies were concatenated and the component types were identified using the random forest or CNN with fully-connected layers. Since 3D coordinates were encoded in features extracted from depth images and PointNet, two types of combinations were evaluated. One is the integration of features from RGB images, intensity images, and depth images, and the other is the integration of features from RGB images, intensity images, and PointNet. Table 3 shows the experimental results. In all cases, the integrated classifiers have achieved better scores than the single-input classifiers. Among the integrated classifiers, the classifier using depth images was better than the one using PointNet. Comparing the random forest and CNN with fully-connected layers, the scores were almost the same. This result means that the integrated classifier can be simply designed as a multi-input classifier using fully-connected layers.

We also evaluate the effectiveness of feature selection. In our method, features extracted from CNN models were reduced according to their effectiveness. Table 4 shows the numbers of effective features extracted from single-input classifiers. Table 5 shows comparison between classification using all features and classification only using effective features. The result shows that effective features were effective for improving the recognition scores.

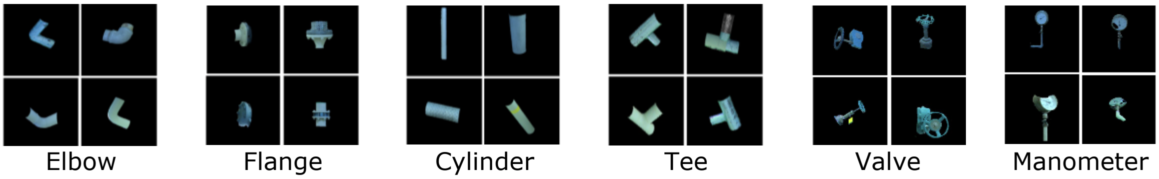


Figure 8: Examples of components.

	Elbow	Flange	Straight	Tee	Valve	Manometer
Point-clouds	171	161	80	21	43	7
Depth Images by CAD models	60	120 (3 types)	80	80	80	80

Table1: Numbers of Data.

When applying this method to actual point clouds, it is necessary to consider components that are not trained for classifiers. We applied our method to points shown in Figure 4. To eliminate unlearned components, we accepted the recognition result only if the classifier output the result with a probability of 70% or more. Otherwise, the result was rejected. Figure 9 shows the recognition results, in which straight pipes, flanges, elbows, and valves are shown in magenta, red, yellow, and green, respectively. Most component types could be successfully recognized, but some distant components could not be correctly identified. This result indicates that the classifier requires a sufficient number of points on each component. In addition, some components could not be correctly separated in the pre-process for classification. It is future work to develop point-cloud segmentation methods based on classification.

	RGB	Intensity	Depth	Depth (+CAD)	PointNet
Elbow	89.8 %	89.8 %	93.8 %	92.0 %	87.8 %
Flange	91.8 %	89.9 %	93.0 %	92.9 %	66.7 %
Straight	93.5 %	95.0 %	92.1 %	95.1 %	77.1 %
Tee	50.0 %	40.0 %	55.6 %	66.7 %	42.9 %
Valve	95.0 %	85.7 %	81.8 %	86.5 %	32.9 %
Manometer	80.0 %	80.0 %	75.0 %	85.7 %	16.7 %
Weighted Average	89.8 %	88.1 %	90.1 %	91.2 %	71.5 %

Table 2: F-measures of the single-input classifiers.

	Random Forest		CNN with Fully-Connected Layers	
	RGB Intensity Depth(+CAD)	RGB Intensity PointNet	RGB Intensity Depth(+CAD)	RGB Intensity PointNet
Elbow	96.5	93.7	95.9	94.5
Flange	96.3	96.3	97.6	93.8
Straight	96.2	95.1	97.6	96.3
Tee	73.7	46.2	66.7	83.3
Valve	95.0	90.5	91.9	83.3
Manometer	100.0	50.0	100.0	75.0
Weighted Average	95.3	92.0	95.2	92.9

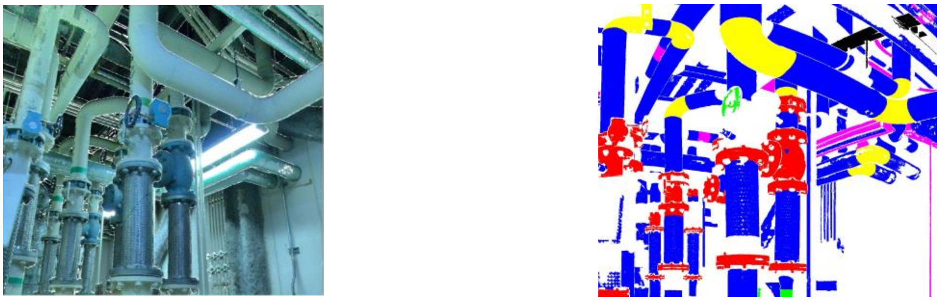
Table 3: F-measures of the integrated classifiers.

	RGB Image	Intensity Image	Depth Image	PointNet
All Features	1024	1024	1024	256
Effective features	602	585	589	216

Table 4: Numbers of effective features.

	All Features	Effective Features
Elbow	95.3 %	96.5 %
Flange	97.5 %	96.3 %
Straight	96.2 %	96.2 %
Tee	60.0 %	73.7 %
Valve	93.0 %	95.0 %
Manometer	100.0 %	100.0 %
Weighted Average	94.6 %	95.3 %

Table 5: F-measures using all features and effective features.



(a) (b)

Figure 9: Classification results: (a) Point cloud, and (b) Classification result.

5 CONCLUSION

In this paper, we proposed methods for identifying components in industrial plants. Since it is difficult to obtain complete point clouds of industrial plants using the terrestrial laser scanner, we discussed methods suitable for only partially acquired points of components. We generated an RGB image, an intensity image, and a depth image from points of each component. The depth images were augmented using CAD models. We trained single-input classifiers, and created the integrated classifiers by concatenating features from single-input classifiers. In our experimental results, the integrated classifier composed of RGB images, intensity images, and augmented depth images was the best among the nine evaluated classifiers. The classification score using the random forest was almost the same as that using CNN with fully-connected layers. In future work, we would like to develop methods for robustly reconstructing 3D models using classification results. We consider methods based on flexible shape templates. 3D catalogs would be useful for selecting a candidate from 3D catalogs and fitting it to points. The challenge is to develop a generic method that can be applied to various types of components. Deep learning could be used for selecting candidates. We would also like to consider methods to handle low-density components. By unifying several low-density components, point density might be increased. In addition, since CNN-based methods have been developed for segmentation, we would like improve segmentation methods for point clouds of industrial plants.

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

REFERENCES

[1] Aurelien, B.; Raphaelle, C.; Raphael, M.; Guillaume, T.; Samir, A.: 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.

[2] Charles, R.-Q.; Hao, S.; Kaichun, M.; Leonidas, J.-G.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Conference on Computer Vision and Pattern Recognition, 2017, 652-660, arXiv:1612.00593.

[3] Charles, R.-Q.; Li, Y.; Hao, S.; Leonidas, J.-G.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Conference on Neural Information Processing System, 2017, arXiv:1706.02413.

[4] Charles R.-Q.; Or, L.; Kaiming, H.; Leonidas, J.-G.: Deep Hough Voting for 3D Object Detection in Point Clouds, Int. Conf. on Computer Vision, 2019, 9277-9286, arXiv:1904.09664.

[5] Charles, R.-Q.; Wei, L.; Chenxia, W.; Hao, S.; Leonidas, J.-G.: Frustrum PointNets for 3D Object Detection from RGB-D Data, Conference on Computer Vision and Pattern Recognition, 2018, 918-927, arXiv:1711.08488.

[6] Fatih, C.; Gabriel, T.: 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.

[7] Hang, S.; Subhransu, M.; Evangelos, K.; Erik, L.: Multi-view Convolutional Neural Networks for 3D Shape Recognition, Int. Conf. on Computer Vision, 2015, 945-953, arXiv:1505.00880.

[8] Hyojoo, S.; Kim, C.; Cgangwan, K.: Automatic 3D Reconstruction of As-Built Pipeline Based on Curvature Computations from Laser-Scanned Data, Construction Research Congress, 2014, 925-934, https://doi.org/10.1061/9780784413517.095.

[9] Jia, D.; Wei, D.; Richard, S.; Li, J.-L.; Kai, L.; Li, F.: ImageNet: A Large-Scale Hierarchical Image Database, Institute of Electrical and Electronics Engineers, 2009, https://doi.org/10.1109/CVPRW.2009.520648.

[10] Jian, C.; Hung, L.-C.; Xiangyu, W.; Chiangzhi, W.; Kwang, H.-J.; Jae, M.-L.: 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.

[11] Joohyuk, L.; Hyojoo, S.; Changmin, K.; Changwan, K.: 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.

[12] Karen, S.; Andrew, Z.: Very Deep Convolutional Networks for Large-Scale Image Recognition, Eighth International Conference on Learning Representations, 2015, arXiv:1409.1556.

[13] Kazuaki, K.; Satoshi, K.; Hiroaki, D.: 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.

[14] Kevin, L.; Liefeng, B.; Xiaofeng, R.; Dieter, F: Sparse distance learning for object recognition combining RGB and depth information, Institute of Electrical and Electronics Engineers, 2011, https://doi.org/10.1109/ICRA.2011.5980377.

[15] Kursa, M., Rudnicki, W.; Feature Selection with the Boruta Package, Journal of Statistical Software, 36(11), 2010, http://dx.doi.org/10.18637/jss.v036.i11.

[16] Masuda, H.; Ichiro, T.; Masakazu, E.: Reliable Surface Extraction from Point-Clouds using Scanner-Dependent Parameters, Computer-Aided Design and Applications, 10(2), August 2013, 265-277, https://doi.org/10.3722/cadaps.2013.265-277.

[17] Masuda, 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.

[18] Masuda, H.; Tanaka, I.: Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds, Computer-Aided Design and Applications, 6(3), 2009, 387-398, https://doi.org/10.3722/cadaps.2009.387-398.

[19] Michael, K.; Matthew, B.; Hugues, H.: Poisson Surface Reconstruction, Eurographics Symposium on Geometry Processing, 2006, 61-70, https://dl.acm.org/doi10.5555/1281957.1281965.

[20] Pierre, A.; David, C.-S.; Y, T.; Mathieu, D.: 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.

[21] Reinhard, K.; Roland, W.; Ruwen, S.; Efficient RANSAC for Point-Cloud Shape Detection, Computer Graphics Forum, 26(2), 2007, 214-226 https://doi.org/10.1111/j.1467-8659.2007.01016.x.

[22] Tomohiro, H.; Tomokazu, K.; Yoshikazu, K.; Kenji, S.: 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.

[23] Zirong, W.; Shuran, S.; Aditya, K.; Fisher, Y.; Linguang, Z.; Xiaoou, T.; Jianxiaon, X.: 3D ShapeNets: A Deep Representation for Volumetric Shapes, Institute of Electrical and Electronics Engineers, 2015, https://doi.org/10.1109/CVPR31182.2015.