www.cadanda.com Computer-Aided Design AND APPLICATIONS

Point Cloud Segmentation for Pipelines in Industrial Facilities Using Recurrent Networks

Kohei Shigeta[1], Takuma Nagumo[2] and Hiroshi Masuda[3]

[1]The University of Electro-communications, k.shigeta.uec@mail.uec.jp
[2]The University of Electro-communications, takuma.nagumo@uec.ac.jp
[3]The University of Electro-communications, h.masuda@uec.ac.jp

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

Abstract. Old facilities in need of renovation often do not have reliable 3D models or 2D drawings because they have been repeatedly modified in their long lifecycles. In such cases, it is effective to measure the facilities using the terrestrial laser scanner and create the 3D model from point clouds. In this paper, we discuss methods for segmenting a pipeline system into each component using a recurrent neural network (RNN). A pipeline is a very long object consisting of various components connected in sequence. For such objects, it is reasonable to encode the pipeline shape in the order of the pipeline direction using the RNN. We introduce an RNN model based on RSNet for segmenting point clouds of pipelines. However, it is generally difficult to prepare a sufficient number of actual point clouds for pipeline systems. Therefore, we propose a method for automatically creating CAD models of virtual pipelines and generating virtual point clouds for training the RNN model. We also propose a method to augment virtual point clouds by simulating occlusions, outliers, noise, unmeasured areas, and non-uniform density. We evaluated our method using point clouds of actual pipelines. The results showed that the proposed method could detect pipeline components from point clouds of industrial facilities with sufficient accuracy.

Keywords: Point cloud, Terrestrial laser scanner, Reverse Engineering, Semantic segmentation, Recurrent Neural Network.
DOI: https://doi.org/10.14733/cadaps.2023.786-796

1 INTRODUCTION

3D models of industrial plants are very effective in planning the layout of equipment installation and simulating maintenance operations such as equipment replacement. Particularly in industrial plants where pipelines are interconnected in complex, it is important to simulate with 3D models in advance because interference between equipment is likely to occur. However, old plants in need of renovation often do not have reliable 3D models or 2D drawings because they have been repeatedly modified in their long lifecycles. In such cases, it is necessary to measure the industrial plant and create the

3D model that is faithful to the existing situation. In recent years, rapid advances in terrestrial laser scanners (TLS) have made it possible to capture dense point clouds from industrial plants.

Since there are many components in industrial plants, it is required to identify and extract each component from point clouds for creating 3D models. In addition, industrial plants often contain many components densely and intricately arranged. Since the laser scanner can only be installed in limited locations, it is very difficult to obtain a complete point cloud from each object. Shape reconstruction of industrial plants requires the creation of 3D models from an incomplete point cloud.

So far, many methods have been proposed to create 3D models of objects from point clouds [1], [2]. Several researchers have tackled the problem of creating 3D models from incomplete point clouds [2][4][14]. Son et al.[14] created skeletons of pipelines by calculating the curvature at each point. Chai et al.[4] created 3D models by matching a point cloud with a library of CAD models using a support vector machine (SVM). Many other methods extract surfaces from point clouds and select an adequate component from industrial standards [6] [8] [9] [11]. Most components in industrial plants consist of primitive surfaces such as planes, cylinders, cones, spheres, and tori. Therefore, the components can be detected by extracting primitive surfaces from the point cloud. It has been reported that planes, cylinders, and spheres can be stably extracted even from incomplete point clouds [8], [9]. Other surfaces, such as tori, can also be detected using the connection relationship for pipelines [6], [11]. For example, the shapes of elbows and tees can be uniquely determined according to industrial standards if the diameter of the connected pipe is given. However, when the shapes of components are not defined in industrial standards, such as the valves and manometers in Figure 1(a), their 3D shapes cannot be uniquely determined from adjacent components. In addition, some components, such as valves and flanges, may have multiple shapes depending on how the components are assembled and combined, as shown in Figure 1(b). Therefore, it is difficult to detect all components only in actual industrial plants using the connection relationships.

Supervised machine learning is expected as a flexible method that can be applied to a wide variety of component shapes. Recently, many convolutional neural network (CNN) methods such as PointNet[15] have been proposed for classification and segmentation of point clouds. However, there are two problems in CNN models for point clouds. First, unlike images, point clouds are not structured and the adjacency between points is not defined explicitly. Second, CNN models are applied to relatively small point clouds, such as 1024 or 2048 points. In order to apply them to tens of millions points, the point clouds must be divided and reduced into a large number of small segments.

Many CNN methods, such as PointNet, performs 1D convolution for a point cloud without using adjacency relationships. PointNet++[16], a derivative of PointNet, determines the representative point from the input point cloud by farthest point sampling, and then performs convolution on gathered k-nearest points. For involving neighborhood relationships, graph-based CNN methods[17] and voxel-based methods[18] have been proposed. The graph-based method creates a graph structure from a point cloud and performs convolution along the paths of the graph. The voxel-based method creates voxels from a point cloud and performs sparse convolution. However, it is difficult for these methods to encode detail shapes in a large-scale point cloud.

In this paper, we discuss methods for segmenting a pipeline system into each component. A pipeline is a very long object consisting of various components connected in sequence. For such objects, a neural network model that encodes the order of points in the pipeline direction would be effective. In CNN models, recurrent neural networks (RNNs) are known to encode sequential data. We regard a point cloud of a pipeline as sequential data and segment the pipeline point cloud into each component using an RNN. We also discuss a method to train the RNN model using automatically created 3D models of pipeline systems.

2 RECURRENT NEURAL NETWORK FOR POINT CLOUDS OF PIPELINES

We consider a CNN model suitable for pipelines. Since a pipeline consists of sequentially connected components, RNN models can be applied to a pipeline point cloud by regarding the pipeline as the time-series data. So far, RSNet [5] has been proposed as an RNN model for point clouds. In this model, the point cloud of the entire scene is divided into rectangular blocks along each of the x, y, and z axes, and 4096 points are selected from each subdivided point cloud. From the selected points, feature values are calculated using 1D convolution. The features in the x, y, and z directions are considered as time-series data, and a bi-directional gated recurrent unit (GRU) is applied to calculate features that include the previous and next data. Finally, the features are used to classify each point. In this research, we consider an RNN model based on RSNet.

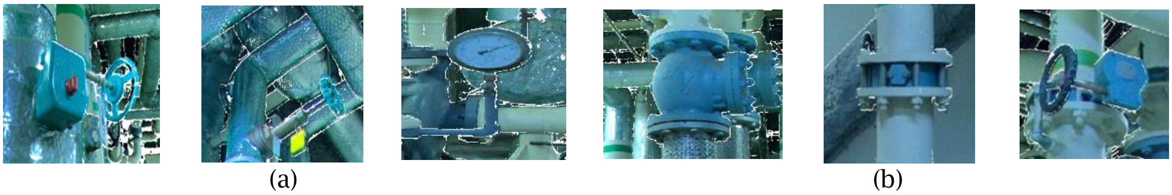


Figure 1: Components of various shapes: (a) non-standard components (valve, manometer), and (b) components created by combining components (flange, torus).

The route of a pipeline can be defined as the central axis. Pipeline components, such as straight pipes and flanges, are arranged along the central axis. Therefore, the point cloud of a pipeline can be regarded as time series data by slicing the point cloud along the central axis. Many methods have been proposed to calculate the central axis of the point cloud of a generalized cylinder [7], [9], [12], [18], [19]. The central axis of a pipeline can be obtained by using these methods.

To handle the pipeline as the time-series data, we define a moving frame with axes e_1, e_2, and e_3 along the central axis, as shown in Figure 2. The point cloud is then sliced along each axis of the moving frame. The features of sequential sliced data are treated as time series data. In our method, the features are computed using the same layers as in the RSNet. Figure 2 shows an overview of our method. To slice the point cloud of a pipeline along the central axis, the moving frame is defined so that the origin is on the central axis and the e_3 axis is aligned with the orientation of the central axis. The axes e_1 and e_2 are orthogonal to e_3. The direction of e_1 and e_2 are arbitrarily determined at the starting point of the pipeline. As the origin of the moving frame moves along the central axis, the e_1 and e_2 axes follow with minimal rotation around the e_3 axis. Coordinates of each sliced point cloud is represented in the moving frame coordinate system, and the coordinates are input to the RNN model. As with RSNet, blocks are created at regular intervals along the central axis, and 4096 points from each block are selected using farthest point sampling. Figure 3 shows the sliced points along the three axes of the moving frame.

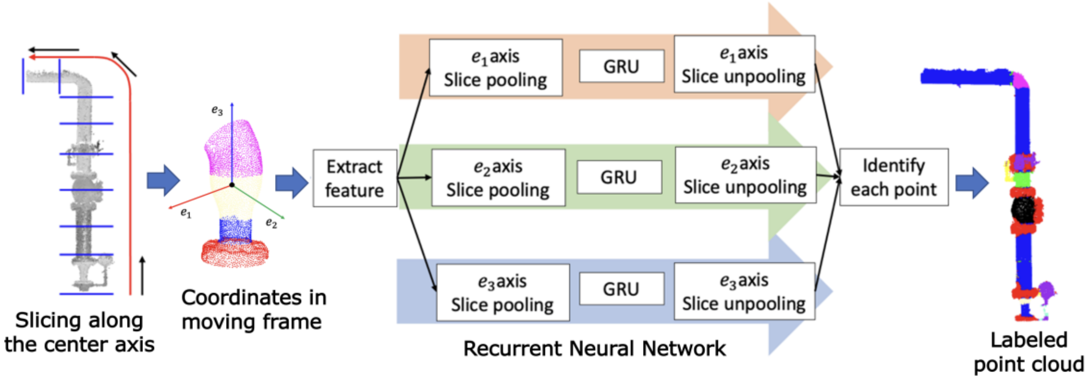
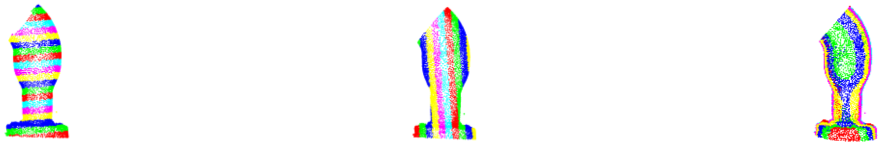


Figure 2: Process for point cloud segmentation.



(a) (b) (c)

Figure 3: Sliced point clouds: (a) Sliced along the $\mathbf{e}_3$ axis, (b) Sliced along the $\mathbf{e}_1$axis, and (c) Sliced along the $\mathbf{e}_2$ axis.

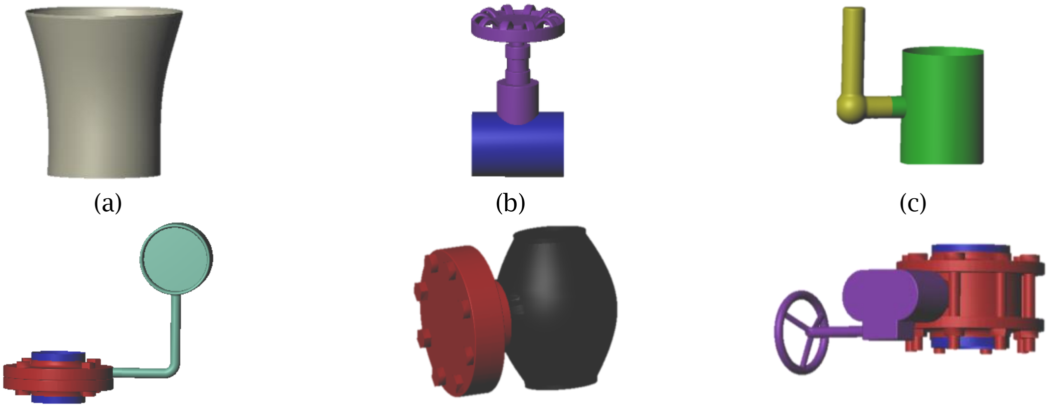
3 TRAINING RNN MODEL

3.1 Creating CAD Models of Virtual Pipelines

To train an RNN model, a large number of pipeline point clouds are required. However, it is difficult to obtain such point clouds by measuring actual pipelines. Therefore, in this research, 3D CAD models of pipelines are automatically generated by randomly connecting components, and point clouds are generated from the CAD models of pipelines.

3D models of pipelines are created by assembling CAD models of components. First the CAD model of each component that appears in pipeline systems are created. If the shape of a component is determined by industrial standards, such as straights, elbows, and tees, the CAD model is created according to industrial standards. On the other hand, components such as reducers, valves and manometers are not defined by industrial standards. In such cases, CAD models are created using point clouds measured in actual industrial plants, as shown in Figure 4(a-c). In addition, we observed that composite components were often used in actual pipelines. In such cases, CAD models of composite components are created by assembling multiple components, as shown in Figure 4(d-f). Composite CAD models are also used for training the RNN models.

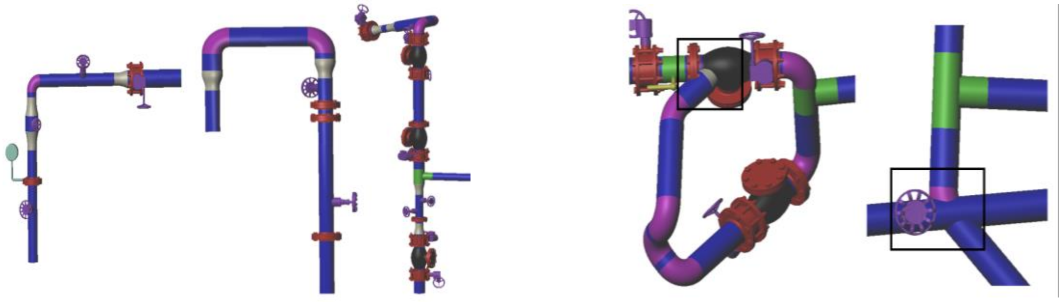
Virtual pipelines are created by randomly selecting and connecting components. For each CAD model, positions and directions for connection are specified as attributes. First, one component is selected, and then other component is selected and connected to the previous component. By repeating this process an arbitrary number of times, virtual pipelines with various shapes can be created, as shown in Figure 5(a).



(a) (b) (c)

(d) (e) (f)

Figure 4: Examples of CAD models: (a) reducer, (b) valve, (c) handle, (d) flange and manometer, (e) flange and ellipsoid valve, and (f) flange and valve.



(a) (b)

Figure 5: CAD models of virtual pipelines: (a) Virtual pipelines and (b) Self-intersections.

However, randomly selecting and connecting components may cause self-intersections, as shown in Figure 5(b). The self-intersections can be easily detected by calculating distances among the center axes of components. If a newly added component intersects with other component, the piping route is changed by inserting or replacing an elbow or a tee to avoid interference.

3.2 Generating a Point Cloud from CAD Model

The terrestrial laser scanner (TLS) measures the coordinate at the point where the laser beam is irradiated. Therefore, a point cloud can be generated from the 3D model of each pipeline if the position and resolution of the virtual TLS is specified. In our method, each CAD model in a virtual pipeline is converted into a dense point cloud, and the point cloud is projected on the 2D angle space defined by the directions of laser beams. This method enables to efficiently simulate point clouds captured at various scanner positions.

First, each CAD model is triangulated. Then, one of triangular faces are randomly selected with a probability proportional to their area. Point P is generated on the triangle using the vertices *A*, *B*, and *C* of the triangle and two random values α and β of [0,1], as follows.

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

By repeating this process many times, a dense point cloud can be obtained. Figure 6 shows a point cloud generated from CAD models in a pipeline. The same label is added to the points generated from each CAD model.

To simulate laser scanning by the TLS, points of each CAD model are mapped on a 2D lattice, which is defined by the angles of laser beams. The interval of the lattice is the angle resolution of laser scanning. The coordinate of each point is converted to (d,θ,ϕ) on the spherical coordinate system with the scanner position as the origin, and the point is projected at (θ,ϕ) on the 2D lattice. Angles θ and ϕ are the elevation and azimuth angles of the laser beam, and d is the distance from the laser scanner. When multiple points are projected on the same pixel, the point with the smallest d is selected. Figure 6(b) shows a point cloud generated by projecting points in Figure 6(a) on the 2D lattice.

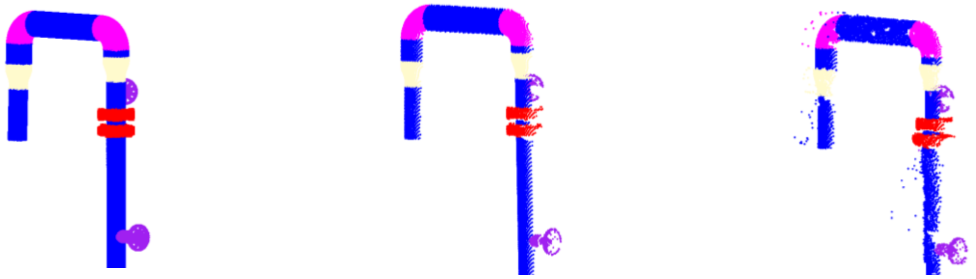
3.3 Data Augmentation for Virtual Point Clouds

Point clouds generated from CAD models are too clean compared to point clouds of actual pipelines. Actual point clouds may be partly missing due to occlusions and contain noises and outliers. In industrial plants, point clouds are measured at multiple locations, and they are unified by registration. Therefore, actual point clouds may contain unmeasured areas or areas with non-uniform point density. In data augmentation, these features are added to point clouds generated from CAD models.

To simulate occlusions, a portion of the point cloud is removed using the region growing method. This process is performed on the 2D lattice defined by (θ,ϕ). First, a seed point is randomly selected and removed. Then, a point is randomly selected from the adjacent points of the removed area. This process is repeated a specified number of times. As a result, a missing portion with a random shape is generated as an occluded region.

Next, outliers caused by the split laser spots are simulated. Outliers are produced if a laser spot is split at the edges of measurable regions, as shown in Figure 7(a). By removing outliers, points become sparse at the edges. To simulate such outliers, the points are removed with a certain probability if the angle between the laser irradiation direction and the surface normal is nearly perpendicular. In this research, we removed points by setting the angle threshold as 65 degree and the probability as 40%. Figure 7(c) shows a point cloud in which points at edges are reduced. In Figure 6(b), points at edges are removed from the point cloud of a pipeline.

We also add outliers to edge points with a small probability, as shown in Figure 6(c) and Figure 8(d). In the laser scanning by the TLS, the coordinates tend to have uncertainty in the direction of laser irradiation. Therefore, we add the normal distribution errors for point clouds in the directions of laser irradiation.



(a) (b) (c)

Figure 6 Augmentation of point cloud generated from a pipeline model: (a) the original point cloud, (b) reduction at edges (c) point cloud with occlusion, noise, and outliers.

(a) (b) (c)

Figure 7 Simulating outliers for virtual point clouds. (a) actual point cloud with outliers, (b) the original point cloud generated from CAD model, and (c) point cloud reduced at edges.

(a) (b)

Figure 8 Data augmentation for virtual point clouds: (a) the original point clouds, and (b) augmented point clouds.

Finally, we simulate unmeasured areas and non-uniform point density caused by registration. From a single pipeline model, many virtual point clouds can be generated by changing the scanner positions. By changing the combination of point clouds, various point clouds with different unmeasured areas and point density can be generated.
By using the methods described above, augmented point clouds can be obtained and used for training the RNN model. Figure 6(c) and Figure 8 show point clouds with occlusions, outliers, noise, unmeasured areas, and non-uniform density.

4 EXPERIMENTAL RESULTS

4.1 Point Clouds for Evaluation

We trained the RNN model using three types of training data. These training data are (a) only actual point clouds captured by the laser scanner, (b) only virtual point clouds generated from CAD models, and (c) both actual and virtual point clouds.

A set of point clouds captured using a TLS were divided into the training data and the test data. For validation data, virtual point clouds were used. The numbers of point clouds for training, validation, and test are shown in

Table 1. The RNN model was trained for 100 epochs using each of the three types of data, and we used the number of epochs with the highest accuracy for the validation data.

In our evaluation, the input point cloud was cut 0.5 m in the direction of the central axis of each pipeline, and 4096 points were uniformly selected from each block. The coordinates were normalized

Point clouds for test	Point clouds for training		Point clouds for validation
Actual only	Actual	Virtual	Virtual only
8	15	249	41

Table 1: Number of pipelines used for training, validation, and test.

to fit in the unit sphere. The training data were superimposed by 0.2 m when slicing into blocks and augmented with Jitter noise [15].

4.2 Evaluation of Training using CAD Models of Pipelines

First, we verified the effectiveness of training using CAD models of pipelines. In this evaluation, virtual point clouds were not augmented by adding occlusions, outliers, and so on.

Table 2 shows the results. The F-values were used as the evaluation metrics. In this evaluation, that training only using actual point clouds were not sufficient, because there were few components other than straights, elbows, and flanges in actual point clouds. In training using virtual point clouds, the accuracy was improved for all component classes compared to the actual point clouds. This result

	Actual point clouds	Virtual point clouds	Both point clouds
Elbow	39.3%	51.4%	**65.8%**
Flange	55.3%	60.9%	**80.8%**
Straight	73.9%	68.1%	**87.3%**
Tee	0%	16.7%	**27.5%**
Valve (handle)	0%	57.7%	**64.3%**
Valve (ellipsoid)	0%	45.4%	**59.0%**
Reducer	0%	5.1%	**11.0%**
Handle	0%	18.3%	**25.2%**
Manometer	0%	30.6%	10.4%

Table 2: F-measures of classifiers using three types of training data.

indicates that virtual point clouds are effective for training the RNN model. In training using both actual and virtual point clouds, the accuracy was significantly improved compared to other two models. This may be because virtual point clouds did not contain occlusion and perturbations, which can be observed in the actual point clouds.

4.3 Evaluation of Training using Augmented Virtual Point Clouds

Next, we evaluated the effectiveness of augmented virtual points with occlusions, outliers, noise, unmeasured areas, and non-uniform density. In this evaluation, we used both actual and virtual point clouds that achieved the highest score in Table 2. The results are shown in Table 3. The F-values were improved or almost equal in most component classes by augmenting virtual point clouds.

However, the RNN model may overfit to perturbation generated by data augmentation when all virtual points are augmented and the number of actual components is relatively small. Therefore, in the next evaluation, we trained the RNN model by augmenting only half of virtual point clouds. The results are shown in Table 4. The score was significantly improved in most component classes. This may be due to suppression of overfitting to perturbation.

4.4 Comparison

We compared the RNN model with PointNet and PointNet++. The training data was both actual and virtual point clouds, and a half of virtual point clouds were augmented. Table 4 shows the results. The RNN model has achieved better overall accuracy compared to PointNet and PointNet++. Compared to PointNet++, the F-values were improved in six in nine classes.

However, the F-values of tees and reducers were lower than PointNet++. Since the RNN model encodes component shapes along the piping route, the scores tend to decrease when the boundary with straight pipes cannot be clearly identified from the point cloud.

	Not Augmented	Augmented (All)	Augmented (Half)
Elbow	65.8%	64.3%	**67.3%**
Flange	80.8%	84.8%	**87.6%**
Straight	87.3%	88.5%	87.6%
Tee	27.5%	30.0%	16.4%
Valve (handle)	64.3%	64.3%	**73.1%**
Valve (ellipsoid)	59.0%	48.0%	**62.1%**
Reducer	11.0%	20.7%	**23.0%**
Handle	25.2%	42.7%	**53.5%**
Manometer	10.4%	53.3%	**61.5%**

Table 3: F-measures of classifiers using augmented virtual point clouds.

	PointNet	PointNet++	RNN (RSNet)	Concatenation of RNN & PointNet++
Elbow	66.5%	58.7%	67.3%	**69.4%**
Flange	72.4%	81.2%	87.6%	**89.8%**
Straight	87.8%	87.5%	87.6%	**88.9%**
Tee	0%	20.7%	16.4%	19.2%
Valve (handle)	34.6%	46.3%	73.1%	**77.9%**
Valve (ellipsoid)	38.7%	50.2%	62.1%	**73.7%**
Reducer	0%	30.4%	23.0%	28.2%
Handle	0%	4.2%	53.5%	**62.5%**
Manometer	0%	72.5%	61.5%	71.4%

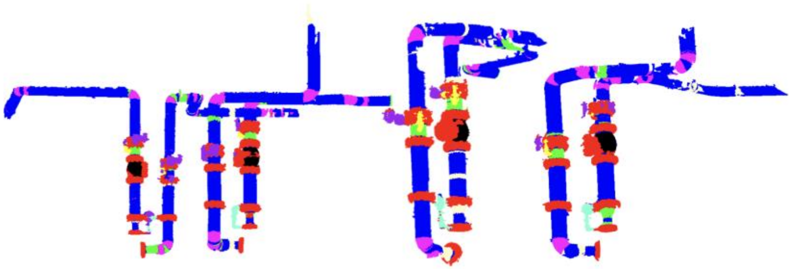
Table 4: Comparison with PointNet and PointNet++.

4.5 Concatenation of Features from Different Models

While the RNN model encodes point clouds along the pipeline route, the PointNet++ model encodes points in k-nearest neighbors. It is considered that these models extract different features from component shapes. Therefore, we created an integrated CNN model trained with features that concatenated the outputs from the RNN and PointNet++ models. From the trained RNN model, we obtained the output of the slice pooling layer, and from the trained PointNet++ model, we obtained the output of the convolutional layer using k-nearest neighbors.

The results are shown in Table 4. For most component classes, the scores were the best among the models in Table 4. As we expected, the integration of different types of shape features is effective for improving the accuracy. However, the accuracy of tees and reducers are still low. This may be because the accuracy was low with both the RNN and PointNet++ models, and both two models could not capture effective features from these component shapes.

Figure 9 shows some results of point cloud segmentation. We applied the integrated CNN model to point clouds of actual pipelines. Points in each point cloud are shown in different colors according to component types. The result shows that our method could detect components in pipeline with sufficient accuracy.



(a)

(b)

Figure 9: Semantic segmentation of point clouds of pipelines: (a)Predicted labels, (b)Ground truth labels.

5 CONCLUSION

This paper described the RNN model for segmenting point clouds of pipelines into each component. Since it is generally difficult to prepare a sufficient number of point clouds for pipelines, we automatically created CAD models of virtual pipelines and generated virtual point clouds for training the RNN model. We also proposed a method for augmenting virtual point clouds by simulating occlusions, outliers, noise, unmeasured areas, and non-uniform density. We evaluated our method using point clouds of pipelines. The results showed that the proposed method could detect pipeline components from point clouds with sufficient accuracy. We also confirmed that the RNN model can be further improved by combining with conventional CNN models that output different shape features.

In future work, we would like to investigate methods for augmenting virtual point clouds to have more features in actual point clouds. In this research, the RNN model was based on RSNet, but other models would be possible for point clouds. We would like to investigate other RNN models. In addition, we would like to investigate CNN models integrated with graph convolution and sparse convolution. In our experiments, the scores of tees and reducers were low. We would like to consider the CNN model that can encode the differences of similar shapes.

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] Huang, W.; Wang, U.: Neumann Recurrent Slice Networks for 3D Segmentation of Point Clouds, Conference on Computer Vision and Pattern Recognition, 2018, 2626-2635. 10.1109/CVPR.2018.00278

[6] 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

[7] 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

[8] 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

[9] Masuda, H.; Tanaka, I.: As-Built 3D Modeling of Large Facilities Based on Interactive Feature Editing, Computer-Aided Design and Applications, Vol.7, No.3, 2010, 349-360. 10.3722/cadaps.2010.349-360

[10] 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

[11] 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

[12] Midorikawa, Y.; Masuda, H.: Extraction of Rotational Surfaces and Generalized Cylinders from Point-Clouds Using Section Curves, International Journal of Automation Technology, Vol.12, No.6, 2018, 901-910. 10.20965/ijat.2018.p0901

[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] 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

[15] 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

[16] 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

[17] Qi, X.; Liao, R.; Jia, J.; Fidler, S.; Urtasun, R.: 3D Graph Neural Networks for RGBD Semantic Segmentation, IEEE International Conference on Computer Vision (ICCV), 2017, 5199-5208. 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

[18] 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. 10.1061/9780784413517.095

[19] Tagliasacchi, A.; Zhang, H.; Cohen-Or, D.: Curve Skeleton Extraction from Incomplete Point Cloud, ACM SIGGRAPH, Vol28, No.3, 2009, 1-9. https://doi.org/10.1145/1531326.1531377

[20] Tchapmi, L. P.; Choy, C. B.; Armeni, I.; Gwak, J. K.; Savarese, S.: SEGCloud: Semantic Segmentation of 3D Point Clouds, 2017 International Conference on 3D Vision (3DV), 2017. 10.1109/3DV.2017.00067