IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 9, 2020, accepted January 27, 2020, date of publication March 5, 2020, date of current version April 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978506

A Convolutional Neural Network for Point Cloud Instance Segmentation in Cluttered Scene Trained by Synthetic Data Without Color

YAJUN XU, SHOGO ARAI, (Member, IEEE), FUYUKI TOKUDA, AND KAZUHIRO KOSUGE, (Fellow, IEEE)
Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan

Corresponding author: Shogo Arai (arai@tohoku.ac.jp)

ABSTRACT 3D Instance segmentation is a fundamental task in computer vision. Effective segmentation plays an important role in robotic tasks, augmented reality, autonomous driving, etc. With the ascendancy of convolutional neural networks in 2D image processing, the use of deep learning methods to segment 3D point clouds receives much attention. A great convergence of training loss often requires a large amount of human-annotated data, while making such a 3D dataset is time-consuming. This paper proposes a method for training convolutional neural networks to predict instance segmentation results using synthetic data. The proposed method is based on the SGPN framework. We replaced the original feature extractor with ''dynamic graph convolutional neural networks'' that learned how to extract local geometric features and proposed a simple and effective loss function, making the network more focused on hard examples. We experimentally proved that the proposed method significantly outperforms the state-of-the-art method in both Stanford 3D Indoor Semantics Dataset and our datasets.

INDEX TERMS Point cloud, instance segmentation, deep learning.

I. INTRODUCTION

Segmentation is an important means to make data easier to understand and analyze. It is helpful for robot tasks [1], autonomous driving [2], augmented reality [3], and visual servoing [4]. Generally, the RGBD image or scene point cloud contains a lot of redundant information, for instance, irrelevant objects and background. It is necessary to grab related details that contain as many key factors as possible and as few irrelevant contents (interference, noise) as possible.

Some progress has been made in point cloud segmentation using deep learning [5]–[8]. In this paper, we focus on bin-picking scenes, where a large number of identical parts are piled up in a box waiting to be aligned. Before pose estimation of parts, an effective segmentation method is beneficial. The segmentation method does not only mitigate computational cost but also improve the precision of pose estimation [9]–[13]. However, these image-based methods are easy to get the defective point cloud by image-based segmentation methods that do not use point cloud directly. For example, Liu *et al.* [11] locate the object in the image using convolutional neural networks (CNN) [14]–[17], and then obtain the corresponding point cloud through bounding box or mask [9]–[11]. Li *et al.* [1] and Li and Hashimoto [13] simply divided the point cloud into many regions of interest (ROI).

Inspired by SGPN [7], which uses a single network for performing instance segmentation on point clouds, we propose a simple and effective method. The proposed method reduces the cost of generating a dataset and improves estimation accuracy. Unlike many current instance segmentation methods based on 2D image [18] or 3D point cloud [19], [20], where RGB image or color information plays an important role, the proposed method requires only point cloud without color. Thus the proposed method can obtain the training data by synthesis. With this method, we can recognize almost all the target objects in the scene and pick out the appropriate point cloud for some robot tasks such as pose estimation and grasping. Note that, although the training dataset is synthetic, the test results were performed on real data.

We made some significant improvements under the framework of SGPN [7], which uses PointNet/PointNet++ [5], [6]

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyu Zhou.

as a feature extractor and predicts instance segmentation based on regression of a similarity matrix, a confidence map, and a semantic prediction. Because PointNet does not show enough performance for the extraction of local information [6], we replace it with the network structure of Dynamic Graph CNN (DGCNN)'s segmentation model [8]. Besides, we propose a focal double-hinge loss function to make the network more focused on the segmentation for difficult cases, as shown in Section III-B.

The main contributions of our works are shown below:

- Performance of instance segmentation by the proposed method outperforms SGPN [7] on the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [21] and our dataset due to the improvements of feature extractor.
- The proposed method requires only point cloud. Thus training datasets can be generated from simulation and easily applied to other objects.
- A novel loss function is proposed and applied to make the training loss easier to converge compared to the loss function of SGPN.
- The proposed Graph CNN trained by synthetic data has excellent performance on real data.
- Experimental results for the proposed method shows the high accuracy of pose estimation of piled-up objects.

The remainder of this paper is organized as follows: Section II introduces some previous progresses and the structure of SGPN. Section III proposes a method of instance segmentation of point cloud. Section IV proves the validity of the proposed method with S3DIS dataset and our dataset for cluttered scene. Section V concludes the paper.

II. RELATED WORK

Object detection and segmentation are core tasks of computer vision. Extracting features from the image by CNN [22], [23] has a better performance than the conventional methods which use hand-crafted features [24]. On the other hand, the past decade has witnessed a rapid increase in the demand for understanding and application of 3D scenes. Pioneers used 3D convolutional neural networks on voxelized shapes [25]. This simple, 3D network-like structure for 3D point clouds, requires high memory requirement and computational cost that limit the practical application.

A. SEGMENTATION ON 2D

FCNs [26] uses a fully connected network (FCN) structure to predict the category of each pixel, which is an end-to-end semantic segmentation network. In SegNet [27], the fully connected network is replaced with the encoder-decoder network, which reduces the computational time and makes predictions more accurate. Since semantic segmentation is unable to distinguish object instances, He *et al.* [18] and Dai *et al.* [28] propose methods of instance segmentation. Based on the work of Faster R-CNN [17], Mask R-CNN [18] adds a subnet to predict the mask of objects. New training data needs to be created to identify new objects, thus the process is time-consuming and laborious. Moreover, data quality can also have an unpredictable effect on prediction results.

B. SEGMENTATION ON POINT CLOUD

In the past few decades, research using point cloud has achieved remarkable achievements. High-quality point cloud scanning technology [29], [30] and breakthroughs in machine learning have strongly promoted the progress in this field. But so far, there are not many methods for point clouds instance segmentation. The methods proposed by Yi *et al.* [19] and Hou *et al.* [20] require not only point cloud but also RGB images. Making such a dataset is resource and time-consuming.

The usual point cloud is unordered, therefore they are invariant to permutations of its members. For the permutation invariance of point clouds, most pioneers prefer to deal with point clouds in a way that handles 2D images, transforming point cloud data to 3D voxel grids [25], [31]. These methods are limited by computational cost. PointNet [5] uses symmetric function to address the permutation invariance of point clouds.

Because PointNet treat each point individually, learn features of each point by multi-layer perception, it does not perform well in the extraction of local information. Wang *et al.* [8] and Qi *et al.* [6] explore the local information of point clouds by searching for the nearest or similar points in embedding space.

Based on PointNet, SGPN [7] proposes a novel method of point cloud instance segmentation. Due to its excellent performance and flexibility, we propose a new method of point cloud instance segmentation based on it.

C. STRUCTURE OF SGPN

SGPN [7] formulates instance segmentation as a clustering problem. It uses PointNet to extract features of point clouds and train a network to predict semantic results, a similarity matrix and point-wise confidence from which instance segmentation results were generated. As shown in Figure 2, SGPN uses a feature extractor to extract features F of size $N_p \times N_f$. N_p is the size of point cloud and N_f is the dimension of feature. F are fed into three subnets, and three feature matrices F_{SIM}, F_{CF}, F_{SEM} are generated with the same shape $N_p \times N_f$. Three subnets are responsible for generating the **similarity matrix**, point-wise **confidence map**, and **semantic segmentation map**, respectively. The loss of SGPN is a combination of three subnets. $L = L_{SIM} + L_{CF} + L_{SEM}$. In brief, SGPN makes the points belonging to the same instance closer in the embedded space, transforming the segmentation problem into a clustering problem.

1) FEATURE EXTRACTOR

SGPN uses the segmentation network of PointNet to extract features of point cloud. Each point corresponds to a

128-dimensional feature, and then these features are sent into three subnetworks for further processing.

2) SIMILARITY MATRIX

In the F_{SIM}, the ith row of feature matrix is a N_f-dimensional tensor representing the position of point P_i in the embedded space. similarity matrix subnet uses the acquired features F_{SIM} to generate a similarity matrix S. The size of S is $N_p \times N_p$. The element S_{ij} in the S is the feature distance of each point pair in the embedded space, which implicitly indicates whether the points P_i and P_j belong to the same instance. SGPN makes the similar point pairs have a close feature distance in the embedded space, although they are far away in the physical space. The relationships of each pair of points P_i, P_j are defined as three classes: 1) P_i and P_j belong to the same instance; 2) P_i and P_j are in the same category but belong to different instances; 3) P_i and P_j do not belong to the same category. Their loss function is:

$$L_{SIM} = \sum_{i}^{N_p}\sum_{j}^{N_p} l(i,j)$$

$$l(i,j) = \begin{cases} \|F_{SIM_i} - F_{SIM_j}\|_2 & C_{ij} = 1 \\ \alpha \max(0, K_1 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 2 \\ \max(0, K_2 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 3 \end{cases} \quad (1)$$

where F_{SIM_i} is the feature of P_i in the embedding space, C_{ij} represents the similarity classes of corresponding point i and j. α, K_1, K_2 are constants. $\alpha > 1$ is to increase the weight of semantic segmentation loss. And $K_1 < K_2$, because the feature distance of point pairs of different categories in embedding space should be greater than that of the same category.

3) CONFIDENCE MAP

The confidence map is a $N_p \times 1$ matrix that indicates how confidently the model thinks the grouping candidate provided by the point is correct. SGPN regresses confidence map based on ground truth groups G as the same form $N_p \times N_p$ as similarity matrix. Each row of the similarity matrix is a cluster proposal: the same instance is less than a certain threshold. For this group proposal, SGPN compares the result of this prediction with ground truth and calculate the intersection over union (IoU). The larger the IoU, the closer the predicted result is to the real cluster, and the more credible the grouping candidate is. The loss L_{CF} between predict group and ground truth group is the L2 loss.

4) SEMANTIC SEGMENTATION MAP

The semantic segmentation map is a point classifier. The number of categories is N_c. SGPN sends F_{SEM} into the subnet and outputs a $N_p \times N_c$ sized matrix M_{SEM}. The element $M_{SEM_{ij}}$ represents the likelihood that the point belongs to each category. The L_{SEM} of the semantic segmentation map is calculated through cross-entropy function.

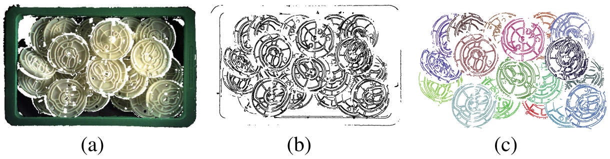


FIGURE 1. Instance segmentation results generated by the proposed method. We first use the same method as [1] to obtain boundary points, and output instance labels for each boundary point. (a) Organized point cloud of Scene. (b) Boundary points in scene cloud. (c) Instance segmentation on boundary point clouds.

III. METHOD

We propose a novel method for instance segmentation on point cloud without color. Figure 1 illustrates the real scene and our instance segmentation result. Inspired by the method of SGPN, which treats instance segmentation as a clustering problem, we directly perform instance segmentation on our boundary point cloud dataset. Boundary points of real scenes are captured by the method proposed in [1]. The boundary points are extracted by performing the Canny Edge algorithm on the RGB image and mapping the corresponding pixels to the point cloud in the scene. We train the network with synthetic data and evaluate our proposed network with real data.

We use DGCNN [8] instead of PointNet as feature extractor of SGPN. DGCNN is a semantic segmentation network with a similar structure to PointNet. The difference is that DGCNN introduces a novel algorithm named EdgeConv [8], which exploit local geometric structures and learn global feature. These features are vital when the synthesized data is missing color information. Besides, for our scenes, we have made some adjustments to the loss function and added a distance mask in the prediction stage. The details are explained in Sec III-B and III-C. Our method is experimentally proven to have better performs on our data set and S3DIS.

A. REPLACE FEATURE EXTRACTOR

The point cloud information we synthesize contains only coordinates, so local features become particularly important. However, PointNet used as the feature extractor in SGPN processes points individually so that the network is unable to learn local features. The problem is caused by the absence of color information because it is difficult to determine the category of a point only by space coordinates. To solve the problem, we first replace feature extractor PointNet with DGCNN and evaluate this modification on S3DIS. We experimentally prove that such a replacement is effective. We report the recognition results of 12 categories (except clutter). The effect of such a replacement plays more important role on our datasets that do not contain color information.

B. FOCAL DOUBLE-HINGE LOSS

SGPN [7] divides the relationship of point pair$\{P_i, P_j\}$ into three classes. Inspired by RetinaNet [32], we make the network more focused on some point pairs, which are on

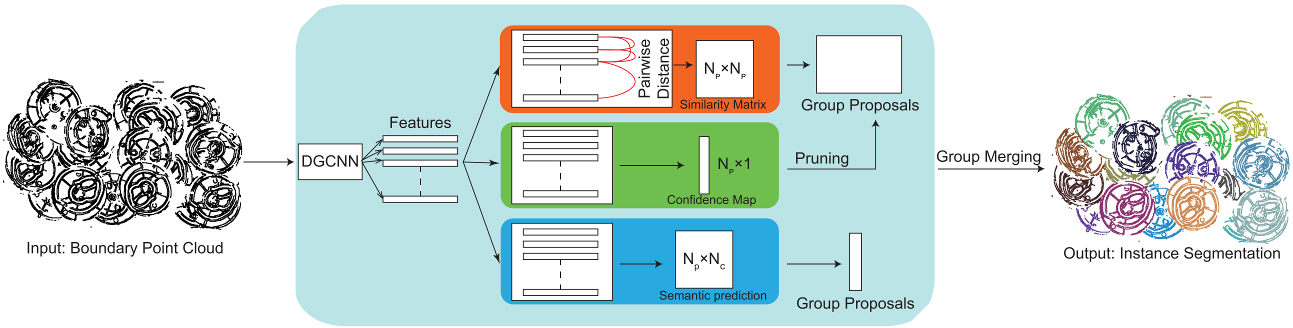


FIGURE 2. Network architecture of modified SGPN. We feed the boundary point cloud into the network and output the instance result for each point. The features of each point are extracted by DGCNN and then sent to three subnetworks respectively. N_p is the size of point cloud, we set $N_p = 4096$. N_c is the number of object categories, in S3DIS $N_c = 13$, in our dataset $N_c = 1$.

the boundary of two different instances. When the feature distance of the point pair is less than a certain threshold, it can be regarded as the same instance. Thus, there is no need for 2 points to contribute to the loss function if they are already close enough. To achieve this, we redefine the loss function as follows:

$$L^*_{SIM} = \sum_i^{N_p}\sum_j^{N_p} l^*(i,j)$$

$$l^*(i,j) = \begin{cases} \beta\max(0, \|F_{SIM_i} - F_{SIM_j}\|_2 - M_1) & C_{ij} = 1 \\ \alpha\max(0, M_2 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 2 \\ \max(0, M_3 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 3 \end{cases} \quad (2)$$

α, β, M_1, M_2 and M_3 are constants such that $\beta > \alpha > 1$, $M_3 > M_2 > M_1 > 0$. Compared to the original loss function, the proposed loss function tends to make the feature distance between two points in the same instance smaller than the threshold M_1 but not need to close to zero. Thus the contribution of easily distinguishable point pairs is down-weighted. Through the improvement, we can find that the prediction results have been greatly improved.

C. DISTANCE-MASK

Inspired by Hinterstoisser *et al.* [12], if the distance d_{ij} between two points is greater than the longest distance $d_{\max}$ in the model, then these two points cannot belong to the same instance as shown in Figure 3. So, we add a distance judgment during the process of clustering points. If their euclidean distance exceeds the maximum size, even if they are very close in embedding space, they will not be regarded as the same instance. We show the improvement effect in Table 2.

IV. EXPERIMENT

We compared the segmentation accuracy of our proposed network with SGPN on S3DIS. The results of SGPN are implemented by the author's code published on GitHub [33]. Scannet Evaluation [34] is adopted to evaluate test results. The predicted instance is considered to be true positive only if the IoU between each predicted and ground truth group is greater than 0.5. An experiment of pose estimation using our dataset is further conducted to prove the validity of segmentation. Our method is implemented in the Tensorflow framework and the hardware devices are an Nvidia GTX1080, Intel Core i7 8700K CPU, and 32G RAM. We use an ADAM [35] optimizer with initial learning rate 0.0001, batch size 2 and momentum 0.9. The network is trained for 200 epochs, which took about 10 hours on each part. During the training phase, $\alpha =$ is set to 2 initially and is increased by 2 every 5 epochs, with a maximum of 10. we set $\beta = 2$ to balance the loss, and $M_1 = 5, M_2 = 10, M_3 = 80$. M_1, M_2, and M_3 are set according to experience and need to satisfy the relationship of $0 < M_1 < M_2 < M_3$. Different values have no obvious influence on the results in our experiment.

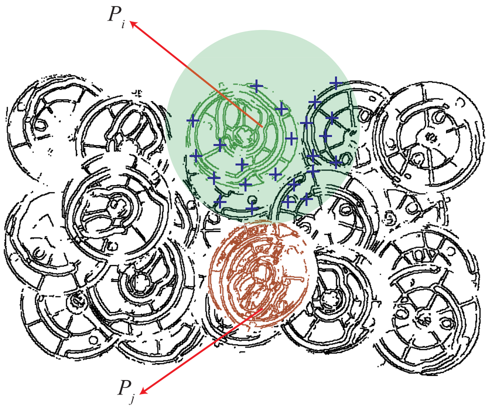


FIGURE 3. The blue circle corresponds to the Distance-Mask of the P_i, and the radius of the circle is the longest distance $d_{\max}$ in the model. Even if P_i and P_j have high similarity in the embedded space, they will not be grouped into one object. Clustering errors can be reduced after adding Distance-Mask.

- S3DIS: The data set involves 3D scans in 6 areas covering 272 rooms. Each point has instance and semantic labels of 13 semantic categories. the network is trained by S3DIS except Area 5 and the network is evaluated by Area 5 of S3DIS. Note that Area 5 did not appear in the rest of the area.
- Our dataset: We use three industrial parts to evaluate the proposed method. Synthesized scenes are generated as the same method as [1], involving 1000 training

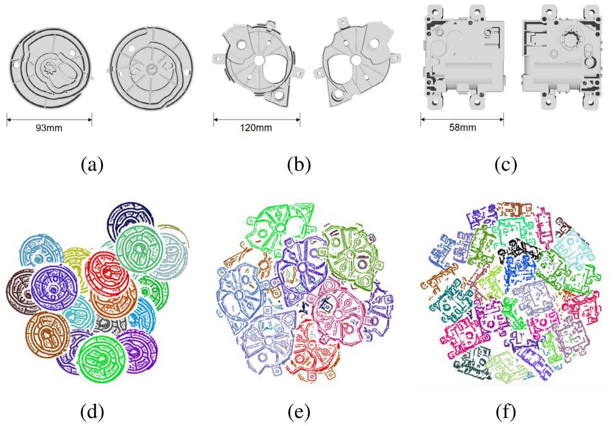


FIGURE 4. (a), (b) and (c) are the models. (d), (e), and (f) are the synthetic boundary point cloud scenes. We visualize the training set and represent different objects in the same scene in different colors.

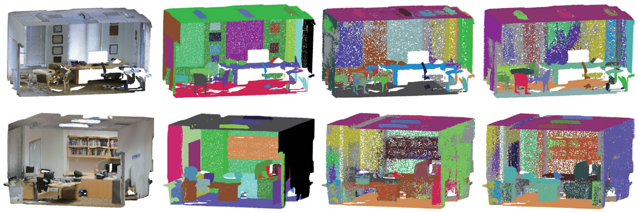


FIGURE 5. Qualitative results of SGPN and ours on the S3DIS. The first col is the real scene, the second col is the ground truth, the third col is the instance segmentation results of SGPN, the last col is the instance segmentation results of ours.

samples. The test samples are real scenes and the ground truth instance labels are made manually. There are 20 to 30 identical types of parts randomly piled up in a scene. Each scene contains about 60,000 boundary points. Each point in the scene has instance annotations. The parts are texture-less and have no discernible color. The models and examples of synthetic scenes are presented in Figure 4. Note that, both of training samples and test samples only contain the boundary points of parts.

A. S3DIS INSTANCE SEGMENTATION

Same as SGPN, we use each point as a 9-Dimension vector (XYZ, RGB, and normalized spatial coordinates). The experimental settings have not changed. Each room is divided into many $1m \times 1m$ blocks, and then 4096 points are sampled. In the test phase, we use all points as the input. Our method is implemented with Tensorflow, Python, and a single GTX1080 GPU. Benefiting from our improvements, Table 1 shows the results on S3DIS, which outperforms SGPN by 4.8-point. The metric is average precision (AP) for each category with an IoU threshold of 0.5. The visualization results are shown in Figure 5.

B. REAL SCENES INSTANCE SEGMENTATION

Figure 6(e) shows instance segmentation results on different real scene. Note that only synthetic data is used during training. Most of the boundary point clouds of container are removed before instance segmentation. Different instances are represented by different colors. We evaluated the segmentation accuracy using two backbones: PointNet and DGCNN. To evaluate our predicted results, we manually make 20 ground truth of Part A. The effect of focal double-hinge loss function and distance mask are reported in Table 2. The metric is AP with IoU threshold of 0.5 and 0.75.

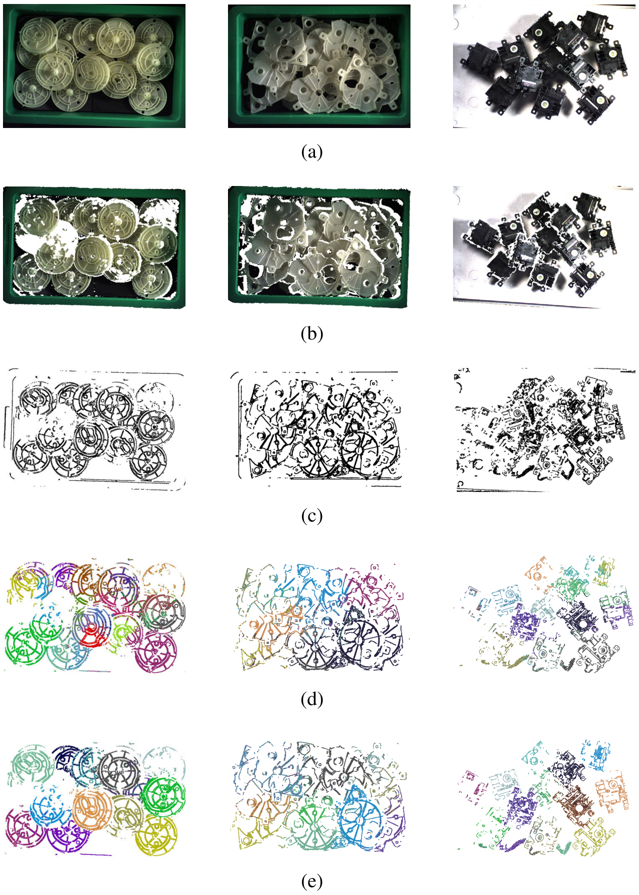


FIGURE 6. (a) is the real scenes. (b) is the organized point cloud of scenes. (c) col is the boundary point cloud of scenes. (d) is the instance results of SGPN. (e) is the instance results of ours.

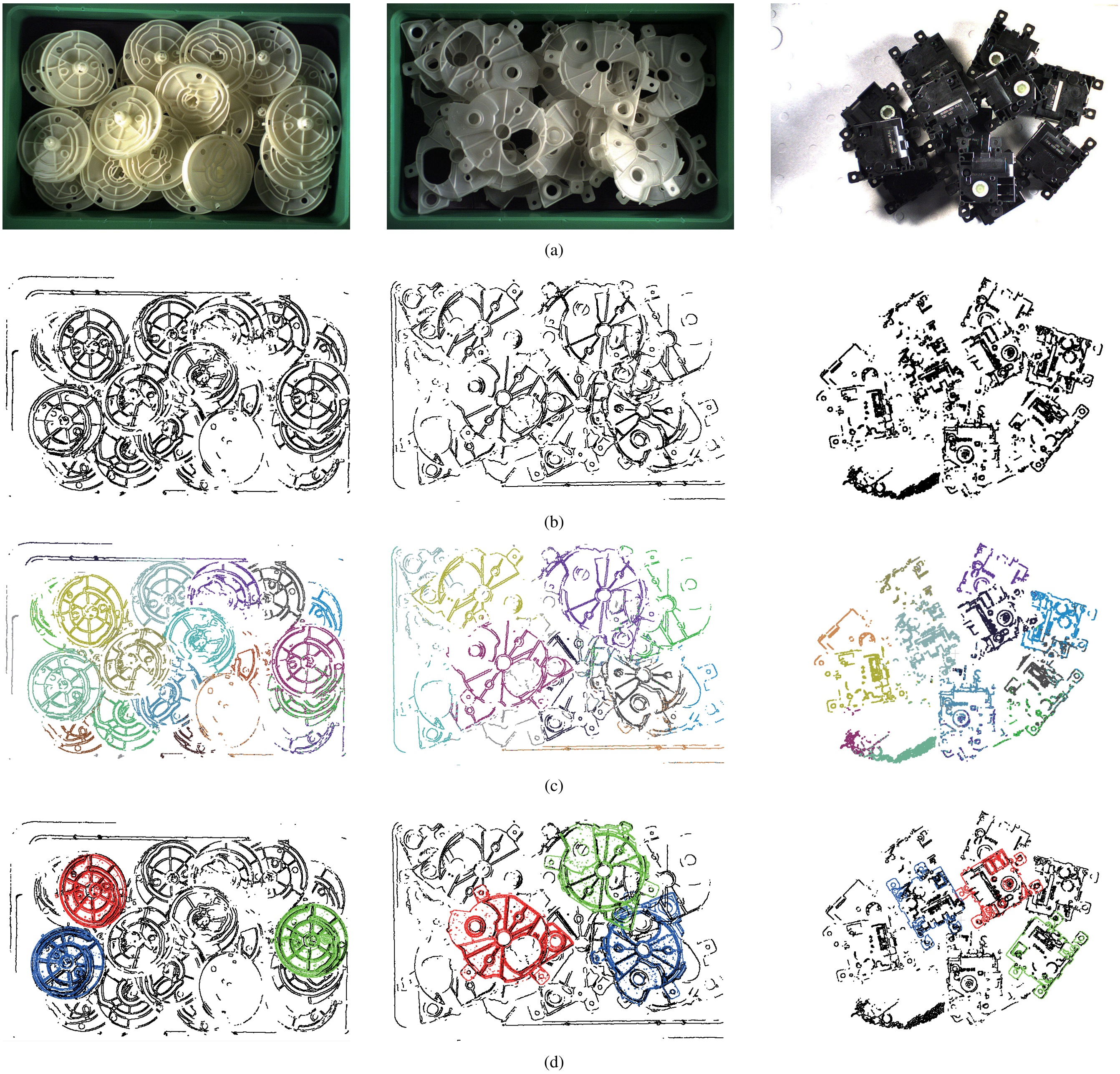
C. POSE ESTIMATION

We test the proposed method in picking scenes where a large number of industrial parts are arranged in a highly unstructured manner in a container. Industrial robots need to measure the 6D pose of the object before handling it. Point pair feature (PPF) based methods or its variants are currently the most effective methods [1], [36]–[39], while pose estimation remains a challenging task with much room for improvement.

We apply the proposed segmentation method to the picking scene. With our segmentation method, the pose estimation method achieves a higher result. The number of points clouds in the model is P, these point groups which contain

TABLE 1. The results of instance segmentation in S3DIS. NL: new loss function.

Method	Backbone	Color	NL	Mean	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board
SGPN	PoinetNet	✓		26.4	43.6	89.4	47.4	0.0	1.0	**55.3**	8.7	17.2	18.6	0.0	**20.5**	15.4
Ours	DGCNN	✓		27.7	44.0	90.5	55.2	0.0	1.3	46.7	2.2	16.0	41.5	0.0	17.5	17.3
Ours	DGCNN		✓	27.6	**48.5**	89.8	50.8	0.0	0.4	32.4	**41.2**	18.3	39.1	0.0	9.2	1.1
Ours	DGCNN	✓	✓	**31.2**	48.2	**91.7**	**62.6**	0.0	**4.0**	42.6	7.7	**22.1**	**53.1**	0.0	13.9	**28.7**


(a)

(b)

(c)

(d)

FIGURE 7. (a) is the real scene. (b) is the boundary point cloud of scene. (d) is the instance results of ours. (e) is the results of the pose estimation. We estimate three results for each scene.

$(1 \pm 10\%)P$ point clouds are selected as candidates for pose estimation from the recognition results of each scene.

PPF-MEAM [1] is used to estimate the pose of parts. PPF-MEAM creates point pair features on the boundary point cloud. Many candidate poses are generated by comparing the point pair features in the model and the real scene. Next, a Hough-like voting scheme is performed to estimate the pose of the part.

TABLE 2. The Results Of instance segmentation in our part a scenes. NL: new loss function, DM: distance-mask.

Method	**Backbone**	**+NL**	**+DM**	$AP_{0.5}$	$AP_{0.75}$
SGPN	PointNet			12.7	0.0
Ours	DGCNN			27.6	3.7
Ours	DGCNN	✓		46.1	23.0
Ours	DGCNN	✓	✓	50.5	25.4

TABLE 3. The results of pose estimation on our bin-picking scenes.

	Part A	**Part B**	**Part C**
Recognition Rate	98.33%	98.33%	93.93%
Mean Segmentation Time on One Scene mean [ms]	2087	1892	1528
Mean Computation Time on PPF-MEAM [ms]	1224	1085	641
Total Mean Time[ms]	3311	2977	2169

We reported the recognition rate in Table 3. The results are shown in Figure 7. Each scene contains approximately 80,000 points, 20 to 30 identical parts with random poses. After segmenting the point clouds in the scene, 3 to 5 groups with the number of points closest to the number of points of model are selected for pose estimation. The estimated pose error can be considered correct if it is within an acceptable range. In our experiment, the metric is $10\% \times d_{\max}$ and 5° in rotation.

V. CONCLUSIONS

This paper proposes a simple and elastic method to segment point cloud at instance-level. We need only the coordinate information of point clouds, so the proposed method is more efficient and convenient, and has overcome the biggest defect of deep learning: dataset. We use S3DIS to confirm that the modification is effective, and the new loss function improves the performance of the model. Experiments show that our algorithm can segment point cloud more precisely than the original instance segmentation method SGPN and has excellent segmentation performance in cluttered point clouds. Furthermore, we conducted a pose estimation experiment using the proposed method and showed that the proposed method could be applied to a precise pose estimation process.

REFERENCES

[1] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, "Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic bin picking," *Sensors*, vol. 18, no. 8, p. 2719, 2018.

[2] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: Towards learning based LiDAR localization for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6389–6398.

[3] P. Speciale, J. L. Schonberger, S. B. Kang, S. N. Sinha, and M. Pollefeys, "Privacy preserving image-based localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5493–5503.

[4] C. Kingkan, S. Ito, S. Arai, T. Nammoto, and K. Hashimoto, "Model-based virtual visual servoing with point cloud data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 5549–5555.

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[7] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578.

[8] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, p. 146, Oct. 2019.

[9] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*. [Online]. Available: http://arxiv.org/abs/1711.00199

[10] O. H. Jafari, S. K. Mustikovela, K. Pertsch, E. Brachmann, and C. Rother, "IPose: Instance-aware 6D pose estimation of partly occluded objects," 2017, *arXiv:1712.01924*. [Online]. Available: http://arxiv.org/abs/1712.01924

[11] D. Liu, S. Arai, Z. Feng, J. Miao, Y. Xu, J. Kinugawa, and K. Kosuge, "2D object localization based point pair feature for pose estimation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 1119–1124.

[12] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 834–848.

[13] M. Li and K. Hashimoto, "Fast and robust pose estimation algorithm for bin picking using point pair feature," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1604–1609.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[15] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[19] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: Generative shape proposal network for 3D instance segmentation in point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3947–3956.

[20] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4421–4430.

[21] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.

[22] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 699–715.

[23] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[24] X. Li, Y. Li, C. Shen, A. Dick, and A. V. D. Hengel, "Contextual hypergraph modeling for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3328–3335.

[25] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[28] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[29] N. Chiba, S. Arai, and K. Hashimoto, "Feedback projection for 3D measurements under complex lighting conditions," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 4649–4656.

[30] B. Schwarz, "LIDAR: Mapping the world in 3D," *Nature Photon.*, vol. 4, no. 7, p. 429, 2010.

[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[33] W. Wang, R. Yu, Q. Huang, and U. Neumann. (2018). *SGPN: Similarity Group Proposal Network for 3D Point Cloud instance Segmentation*. Accessed: Sep. 11, 2019. [Online]. Available: https://github.com/laughtervv/SGPN

[34] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[36] S. Arai, T. Harada, A. Touhei, and K. Hashimoto, "3D measurement with high accuracy and robust estimation for bin picking," *J. Robot. Soc. Jpn.*, vol. 34, no. 4, pp. 261–271, 2016.

[37] W. Abbeloos and T. Goedemé, "Point pair feature based object detection for random bin picking," in *Proc. 13th Conf. Comput. Robot Vis. (CRV)*, Jun. 2016, pp. 432–439.

[38] J. Vidal, C.-Y. Lin, and R. Martí, "6D pose estimation using an improved method based on point pair features," in *Proc. 4th Int. Conf. Control, Automat. Robot. (ICCAR)*, Apr. 2018, pp. 405–409.

[39] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3828–3836.

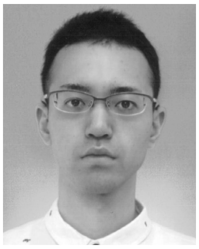
YAJUN XU was born in Ya'an, Sichuan, China, in 1992. He received the B.S. degree in mechanical engineering from Central South University, Changsha, China, in 2015. He is currently pursuing the degree with Tohoku University, Japan.

His research interests include deep learning, 2D/3D segmentation, and 6D pose estimation, especially object detection in the bin-picking scene.

SHOGO ARAI (Member, IEEE) received the B.S. degree in aerospace engineering and the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2005, 2007, and 2010, respectively.

From 2010 to 2016, he was an Assistant Professor with the Intelligent Control Systems Laboratory, Tohoku University. In 2016, he joined the System Robotics Laboratory, Department of Robotics, Tohoku University, as an Associate Professor, where he is currently an Associate Professor. His research focuses on the fields of robot vision, machine vision, 3D measurement, production robotics, networked control systems, and multiagent systems.

Dr. Arai received the 32th Best Paper Award from The Robotics Society of Japan, in 2019, the Certificate of Merit for Best Presentation from The Japan Society of Mechanical Engineers, in 2019, the Excellent Paper Award from The Institute of Systems from Control and Information Engineers, in 2010, the Best Paper Award Finalist at the IEEE International Conference on Mechatronics and Automation, in 2012, the SI2019 Excellent Presentation Award from The Society of Instrument and Control Engineers, in 2019, the SI2018 Excellent Presentation Award from The Society of Instrument and Control Engineers, in 2018, the SI2017 Excellent Presentation Award from The Society of Instrument and Control Engineers, in 2017, and the Graduate School Research Award from the Society of Automotive Engineers of Japan, Inc., in 2007. He received the Best Paper Award from the FA Foundation, in 2019.



FUYUKI TOKUDA received the B.S. degree in engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 2017, and the M.S. degree in engineering from Tohoku University, Sendai, Japan, in 2019, where he is currently pursuing the Ph.D. degree in engineering.

His research interests include the development of visual feedback control using deep learning for bin-picking and assembling of industrial parts.

KAZUHIRO KOSUGE (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in control engineering from the Tokyo Institute of Technology, in 1978, 1980, and 1988, respectively.

From 1980 to 1982, he was a Research Staff with the Production Engineering Department, Nippon Denso Company, Ltd. From 1982 to 1990, he was a Research Associate with the Department of Control Engineering, Tokyo Institute of Technology. From 1990 to 1995, he was an Associate Professor with Nagoya University. Since 1995, he has been with Tohoku University. He is currently a Professor with the Department of Robotics, Tohoku University, Japan.

Dr. Kosuge is a member of the IEEE-Eta Kappa Nu and the IEEE Vice President for Technical Activities for 2019. He received the JSME Awards for the best papers from the Japan Society of Mechanical Engineers, in 2002 and 2005, and the RSJ Award for the best papers from the Robotics Society of Japan, in 2005. He is a JSME Fellow, a SICE Fellow, a RSJ Fellow, and a JSAE Fellow. He was the President of the IEEE Robotics and Automation Society, from 2010 to 2011 and the IEEE Division X Director, from 2015 to 2016.