2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Sim-to-real Object Pose Estimation for Random Bin Picking

Boyoung Kim[1*] and Junhong Min[1]

***Abstract*— In industry, random bin picking is a complex and difficult task where instance segmentation and object pose estimation based on point clouds are key processes. Recently, learning-based segmentation and pose estimation methods for 3D point clouds have been proposed. However, many of them require supervised learning with datasets with annotations of objects. Since it is difficult to annotate all stacked instances in bin picking dataset, learning without real-world datasets has become a major interest. In this paper, we introduce an instance-level object pose estimation method for bin picking, which is trained using only simulated data and seamlessly applied to real-world scenarios without additional adaptation. To enable this, we introduce a method for generating a comprehensive synthetic dataset using a physics simulator, which incorporates 3D CAD models of objects and automatically generates annotations for both segmentation and pose estimation. Our experiments, conducted on synthetic datasets, highlight the competitive performance of our method in terms of recall and accuracy. Furthermore, we demonstrate the successful integration of our approach with real robot random bin picking, resulting in significantly improved picking success rates.**

I. Introduction

Random bin picking has been pivotal in various industrial sectors, such as manufacturing, warehousing, and logistics [1], [2], [3]. This task entails extracting disorganized objects from containers where the orientations and positions of these items are variable and not predefined. Consequently, it involves recognizing the poses of objects randomly placed within the bin. However, despite its importance, this task remains challenging due to the unpredictability of object orientations, diverse stacking configurations, and potential occlusions among objects [4].

Several approaches in computer vision and robotic have addressed this challenge, in terms of multi-object pose estimation. These approaches can be categorized into two primary categories based on the dominant data type: image-based and point cloud-based methods. First, The image-based methods estimates object poses by employing either RGB data exclusively or in conjunction with depth information. Classical methods of these methods include template matching [5], [6] and feature-based [7]. Currently there are image-based pose estimation studies using deep learning[3], [8], [9], [10], [11]. There is also a method using only depth images [12]. However, image-based approaches have limitations including low depth resolution and vulnerability to fluctuations in lighting conditions, making them less suitable for industrial applications. Moreover, it only works in conditions where there is little or no occlusion.

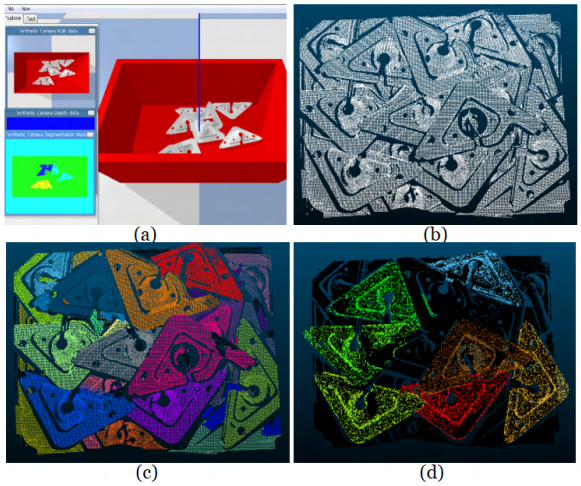


Fig. 1. Object pose estimation result of real data. Point clouds with low reliability are displayed in black. Instance poses are accurately estimated from an actual data through learning only with simulation data. (a) Simulator for bin picking data generation, (b) Actual 3D sensor data, (c) Instance segmentation result, (d) Pose estimation result.

On the contrary, the point cloud-based approach involves the estimation of object poses, harnessing the rich spatial information contained within 3D point cloud data. The advantage of point cloud data lies in its ability to faithfully represent the geometry of objects. Within the realm of classical methods, several notable approaches are employed, including point feature matching [13], [14] and the Hough transform [15], each offering unique strengths and applications. Particularly noteworthy among these methods is Point Pair Feature (PPF) matching [16], which belongs to the broader category of point feature matching. PPF matching has emerged as one of the most prominent and effective means of precisely estimating object poses, capitalizing on the geometric properties of point pairs to robustly determine object orientations.

Learning-based methods have been introduced in the context of 3D point cloud data analysis. Nevertheless, compared to the widespread adoption of deep learning in image-based methods, the integration of learning-based approaches into 3D point cloud-based methods occurred somewhat later. This delay can be attributed to the unstructured and unordered characteristics of point clouds [18], which required the development of neural network architectures capable of efficiently handling such data. With the emergence of pioneers in extracting features from 3D point clouds [19], several deep

[1]Boyoung Kim and Junhong Min are with Global Technology Research, Samsung Electronics, 129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Republic of Korea by1110.kim@samsung.com junhong1.min@samsung.com
* Corresponding Author

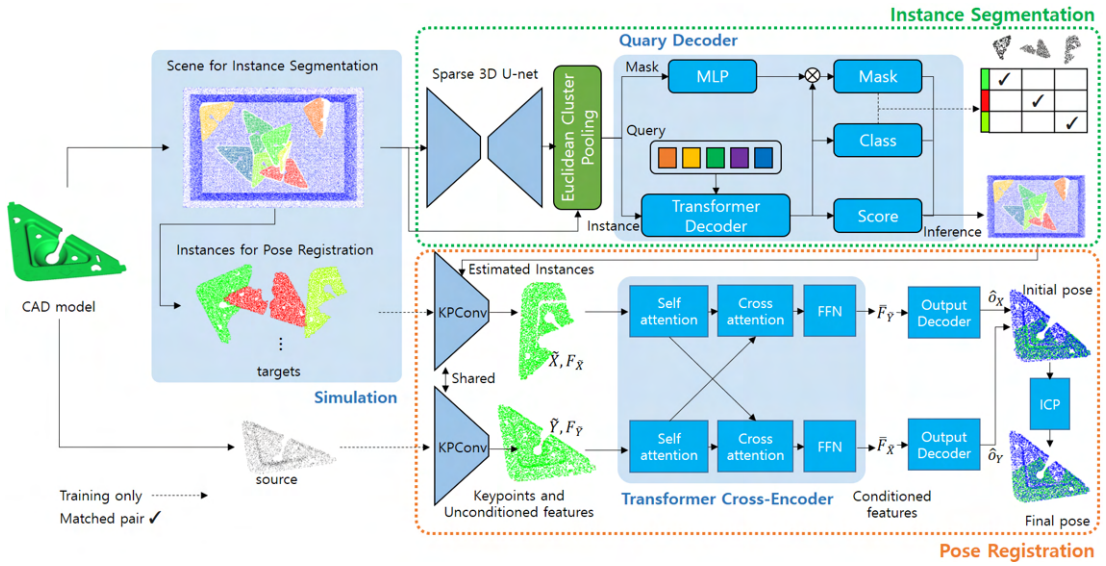


Fig. 2. The overall architecture of sim-to-real pose estimation for bin picking. It consists of three main parts: data generation, segmentation, and pose registration. First, in the data generation process (Sec. III-A), the pose and labeled point cloud data are generated through simulation from the CAD file. In the segmentation part (Sec. III-B), we group the 3D U-Net features from the labeled point cloud data. The transformer-based query decoder enables query vectors to learn instance information through feature cross-attention. Pose estimation network (Sec. III-C) learns the probability of each point lying in the overlapping region and its corresponding position in the other point cloud. Then, an initial pose is calculated using these probabilities. Finally, Iterative Closest Point (ICP) [17] fine-tunes the final pose based on the inferred initial pose.

learning studies have recently emerged to estimate object pose from point cloud data [18], [20], [21]. However, most of them assume that there is annotated data in the form of supervised learning.

The significance of training datasets is pivotal in the domain of learning-based methods. Fortunately, a diverse datasets tailored for training on 3D point cloud data are readily available. However, a majority of these datasets predominantly focus on indoor scenes or are specifically designed for autonomous driving scenarios [22], [23], [24], [25]. In stark contrast to these relatively well-structured environments, the environment in bin picking presents a much greater level of complexity, largely attributed to the difficulties arising from object overlap. Furthermore, the task of annotating 3D point clouds to build high-quality training datasets for bin picking is extremely labor-intensive.

In this paper, we present a novel object pose estimation framework designed specifically for sim-to-real applications in random bin picking scenarios. To tackle the associated challenges, we introduce a two-stage network that breaks down the random bin picking problem into two distinct subproblems: instance segmentation and 3D point cloud registration. Our approach is influenced by established networks in both of these domains, as documented in prior work [26], [27], and we further enhance them by incorporating synthetic data for training. Our method generates a bin picking dataset using a physics simulator, systematically altering the drop locations for objects. This deliberate variation in drop points has led to a wide range of object pose variations in the dataset, as illustrated in Fig. 1 (a). The proposed method accurately captures the characteristics of real-world bin picking scenarios. Furthermore, it exhibits superior recall and precision performance compared to previous studies conducted in simulated environments. Most notably, it demonstrated its ability to seamlessly transfer to real-world bin picking situations.

The main contributions of this work are as follows: 1) A novel bin picking solution is proposed, relying solely on CAD files and consisting of two stages: segmentation and pose estimation; 2) Through experiments on both open datasets and our custom data, we verified superior recall and precision performance compared to previous research; 3) We further performed our sim-to-real method on robot bin picking in real world, verified the adaptability and applicability.

II. Related Work

A. Instance Segmentation in Point Cloud

Instance segmentation method can be divided by proposal-based and grouping-based methods. Proposal-based methods take a top-down pipeline. They estimate region of instance objects after predicting bounding boxs or masks. 3D-BoNet [28] performs a direct regression to estimate 3D bounding boxes for all instances in a point cloud, while also predicting a point-level mask for each instance. GSPN [29] carried on with the concept of Mask R-CNN [30]. From predicted region proposals, GSPN refines the proposals and generates segments. 3D-MPA [31] and PPR-Net [20], [32] use an object-centric approach where each point votes for its object center. Based on center points, clustered point clouds belongs corresponding object instances. However, in the bin picking scene, points exist only on the surface of the object, making it difficult to estimate the center of the object. Moreover,

low-quality candidate regions reduce instance segmentation performance.

Grouping-based methods consider 3D instance segmentation as a bottom-up pipeline. They extract features first and then group point into object instances. Most of the recent studies showing good performance in instance segmentation are grouping-based methods. SGPN [33] proposed a network to predict point grouping proposals and a corresponding semantic class for each proposal. It regards similar features as the points of the same instance. FPCC [18] continued the idea of clustering confidence scores of the points in SGPN. FPCC predicts that an instance takes only one point, which is most likely the geometric center of the object, which is difficult to apply if the length of the object is not similar in all directions. PointGroup [34] aggregates points from original and offset-shifted point clouds and formulates ScoreNet to evaluate aggregation scores. SSTNet [35] acquires instances by building a semantic superpoint tree and splitting distinct nodes. SPFormer [26] grafts the superpoint tree into transformer [36]. Except for FPCC [18], most of verification was conducted on a dataset without occlusion, which is difficult to apply immediately in bin picking with frequent cluttered scenes. Unlike indoor scenes or self-driving scenes, real-world bin picking scenes are a collection of small clutters that are not monotonous due to stacked objects, and a segmentation network that reflects these environments is required.

B. Pose Registration in Point Cloud

Classical methods typically utilized PPF to estimate 6D object pose [16], [37]. They achieves sufficient results when scenes are clear and limited optical environment. However, PPF-based methods are sensitive to noise and occlusion, which are frequent in real world bin-picking scenes [12]. Current studies propose learning based pose estimation for handling complex occlusion scenes [27], [20]. PPR-Net [20] extends learning-based Hough voting to point cloud. REGTR [27] focuses on having the network predict a set of clean correspondences. PPR-Net calculates the occlusion rate in advance and calculates mainly those with high visibility, whereas REGTR robustly learns a set of good correspondents even in high occlusion and utilizes a transformer structure that has recently shown remarkable performance, therefore this study adopted REGTR.

III. Method

Fig. 2 shows the comprehensive framework of our approach, comprising data generation, segmentation network, and pose estimation network components.

A. Data Generation using Simulator

A simulated environment is built with PyBullet [38]. After setting the number of objects to generate, the location to drop them, a CAD file path, and the sensor information, pybullet creates bin picking scenes. Each instance is labeled. In case some point clouds are segmented with incorrect labels, the data for pose estimation is generated by including some incorrectly labeled point clouds as noise. In addition, the ground truth pose is saved for each instances. As a post-processing of the dataset, the field of view information is used to remove point clouds in areas that are not visible. Labeled cluttered scenes and ground-truth poses are leveraged for training in instance segmentation and pose registration.

Simulator data can be used for learning because current 3D sensors, like structured light-based ones, have high resolutions that differ little from simulated datasets to real scenes. The Zivid Two and Photoneo PhoXi S are examples with point precisions of 55 and 50μm respectively.

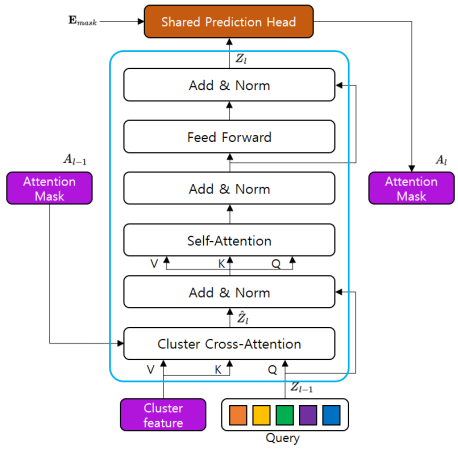


Fig. 3. The scheme of transformer decoder layer.

B. Instance Segmentation

Based on the architecture of SPFormer [26], we modified pooling layer in order to represent delicate features in complex bin picking scene.

1) Feature Extraction & Abstraction: We convert the input point clouds $\mathbf{P} \in \mathbb{R}^{N\times 3}$ into a point-wise features $\mathbf{P}' \in \mathbb{R}^{N\times C}$ using a sparse 3D U-net. Each point has 3D coordinates x, y, z. KD-tree [39] based Euclidean clustering binds those potential features into small clutters. Euclidean cluster pooling layer easily get features $\mathbf{E} \in \mathbb{R}^{M\times C}$ via average pooling over those point-wise ones inside each clusters. Extracting superpoints requires a lot of computation to specify a mid-level shape representation, whereas Euclidean clustering quickly clusters only with distances from neighboring points. Since bin picking scenes are complex, it is more appropriate to quickly collect smaller clusters than to acquire hundreds of mid-level shape clusters. This feature pooling layer significantly reduces the computational overhead of subsequent processing and enables efficient and stable network design.

2) Query Decoder with Transformer: Subsequently, the query decoder with transformers enables K query vectors to learn instance information through feature cross-attention. Query decoder is composed of an instance branch and a mask branch. In the mask branch, the mask-aware features $\mathbf{E}_{mask} \in \mathbb{R}^{M\times D}$ are obtained by a Multi-Layer Perceptron (MLP). The instance branch consists of a series of

TABLE I

INSTANCE SEGMENTATION RESULTS FOR OPEN DATASET. THE METRICS ARE PRECISION (%) AND RECALL (%) WITH AN IOU THRESHOLD OF 0.5.

	Object A		Object B		Object C		Ring Screw		Gear Shaft	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
SGPN [33]	41.9	29.0	20.1	25.9	24.6	25.4	10.9	14.7	15.3	21.7
ASIS [40]	64.7	28.7	55.4	23.6	67.1	42.1	15.5	11.6	20.1	9.5
3D-BoNet [28]	66.1	50.2	42.2	26.4	45.9	62.4	27.9	19.8	26.6	20.1
PointGroup [34]	91.9	46.2	75.2	39.4	**79.8**	41.1	52.4	41.2	58.8	36.8
FPCC [18]	89.7	67.3	**80.9**	50.9	78.6	64.3	58.4	48.7	54.3	69.5
Proposed Method	**96.1**	**97.3**	79.7	**85.3**	74.7	**92.2**	**93.9**	**95.5**	**95.0**	**97.9**

transformer decoder layers. They enables query vectors via feature cross-attention. The features of query vectors from each transformer decoder layer as $\mathbf{Z}_l \in \mathbb{R}^{K\times D}$ are predefined as where D and l are embedding dimension and layer index, respectively. The transformer decoder handles potential features of clusters with arbitrary lengths and learnable query vectors. A specified transformer decoder layer is shown in Fig. 3. Since the decoder receives features as input, the position embedding is removed. Cluster features $\mathbf{E}' \in \mathbb{R}^{M\times D}$ after linear projection, and query vectors Z_{l-1} from the previous layer capture context information via a cluster cross-attention mechanism. The output of the feature cross-attention can be defined as:

$$\hat{Z}_l = softmax(\frac{\mathbf{QK}^T}{\sqrt{D}} + \mathbf{A}_{l-1})\mathbf{V} \tag{1}$$

where $\mathbf{Q} \in \mathbb{R}^{K\times D}$ is the linear projection of query vectors. $\mathbf{K}$, $\mathbf{V}$ is cluster features $\mathbf{E}'$ with different linear projection. Cluster attention masks $\mathbf{A}_{l-1}$ can be formulated as follows.

$$\mathbf{A}_{l-1}(i,j) = \begin{cases} 0 & \mathbf{M}_{l-1}(i,j) \geq \tau \\ -\infty & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbf{M}_{l-1}(i,j)$ shows the predicted cluster masks. i-th query vector attends to j-th cluster when the mask $\mathbf{M}_{l-1}(i,j)$ is higher than τ. Finally, the segmentation network is trained for instance segmentation through bipartite matching based on feature masks. After training, the segmentation network can directly predict instances with classification and corresponding feature masks. The shared prediction head and bipartite matching are the same as those in SPFormer [26], and the details can be found in it.

C. Pose Registration

Excluding the dataset generation, we adopted a pose estimation network from REGTR [27]. Fixed source files and objects with various poses and occlusions are paired to form the dataset. KPConv extracts downsampled keypoints and their associated features from point clouds of CAD and instances. Then, our system passes these keypoints and features to transformer cross-encoder layers. The network for pose estimation learns the probability that each point lies in the overlapping region and its corresponding position in the other point cloud. The corresponding transformed locations of the keypoints in the other point cloud are obtained with overlap scores. An initial pose is computed from the good correspondences. ICP [17] fine-tunes the final pose based on the inferred initial pose.

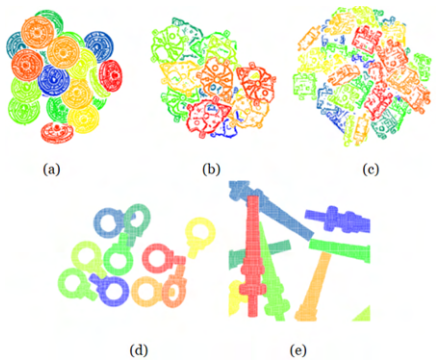


Fig. 4. Synthetic 3D point cloud training scenes for object A, B, C, Ring Screw and Gear Shaft.

IV. EXPERIMENT

We conducted tests on five objects from open and our custom datasets. In the open dataset experiment, we evaluated performance by using real data objects A, B, and C from the XA dataset [41], as well as Ring Screw, Gear Shaft of IPA Bin-Picking dataset [2]. Additionally, we also verified the effectiveness of the modified pooling layer using scenes generated from a single CAD model and real sensor data. Finally, our proposed method successfully estimated 6D poses of stacked objects through simulation training only, executed a robot bin picking task, and evaluated its recall performance.

Our segmentation and pose estimation network are trained using AdamW [42] optimizer, with an initial learning rate of 1e-4 and weight decay of 1e-4 on a single Nvidia RTX 4090 GPU. We set the batch size as 4 for both. We preprocessed our custom data by downsampling to 1 mm voxel units.

A. Benchmark Results

In the XA dataset, there are 500 training scenes and 20 test scenes for each object. There are 17,500 and 15,000 training scenes, and 280 and 250 test scenes for Ring Screw and Gear Shaft, respectively. Fig.4 shows examples of a synthetic training scene for each object. Due to the objects being stacked, there are areas where the objects below are not visible, demonstrating that the scenes for bin picking are accurately replicated.

Table.I illustrates performance measurements using precision and recall, with an IoU threshold of 0.5. The recall

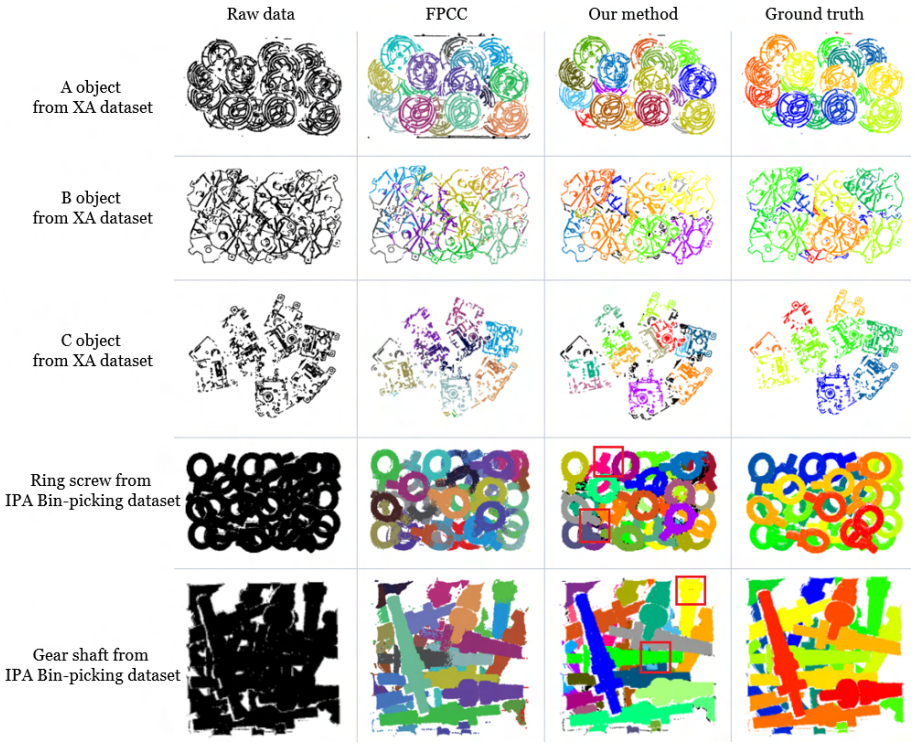
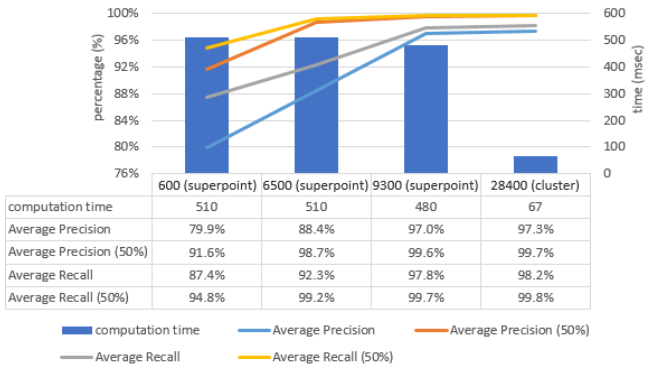


Fig. 5. Visualization results of our instance segmentation and FPCC [18] result on XA and IPA dataset.

TABLE II
COMPUTATION SPEED COMPARISONS (MSEC/SCENE).

	Object A	Object B	Object C
FPCC [18]	117.831	114.197	113.613
Proposed Method	80.637	75.241	53.247



	600 (superpoint)	6500 (superpoint)	9300 (superpoint)	28400 (cluster)
computation time	510	510	480	67
Average Precision	79.9%	88.4%	97.0%	97.3%
Average Precision (50%)	91.6%	98.7%	99.6%	99.7%
Average Recall	87.4%	92.3%	97.8%	98.2%
Average Recall (50%)	94.8%	99.2%	99.7%	99.8%

Fig. 6. Segmentation results based on number of superpoints and clusters.

performance of our proposed method excels across all test objects. In the case of specially shaped point clouds, such as edge-shaped ones, it effectively demonstrates precision by grouping homogeneous points at the semantic level. However, the score difference does not exceed 5%, and in commonly encountered objects such as ring screw and gear shaft, the proposed method exhibits significantly higher precision than FPCC by more than 35%. Table. II reports the average computation time per scene. Our proposed method is faster than FPCC [18], completing tasks in under 100 msec.

The visualized segmentation results are presented in Fig. 5. In the second and third columns, the segmentation results of FPCC [18] and our method are depicted, respectively, with different colors indicating different instances. As observed in the quantitative results, improved segmentation results are evident within the IPA Bin-Picking dataset. The segmentation superiority of our method is highlighted by a red bounding box in Fig. 5.

B. Our Dataset

We use a triangle-shaped part to generate our synthetic and real dataset. For the synthetic data experiment, we generated 1500, 450 and 165 synthetic scenes for training, validation and test sets, respectively. Test scenes comprised of 1320 instances, which were used for pose estimation training purposes. The training time for segmentation and pose registration were approximately 9.5 hours for 180 epochs and 6 hours for 300 epochs, respectively.

1) Ablation on pooling layer: We compared the performance of clustering and superpoint pooling layers as a bridge between the backbone and query decoder, and the results on the synthetic data are shown in Fig. 6. Superpoint pooling layer groups homogeneous neighboring points. While superpoint takes time to identify homogeneous groups, it can be seen that performance improves as the number of superpoints increases (the group size becomes smaller). However, as group sizes become smaller, identifying homogeneous groups becomes less meaningful; therefore, grouping them into a cluster pooling layer can speed up computation and

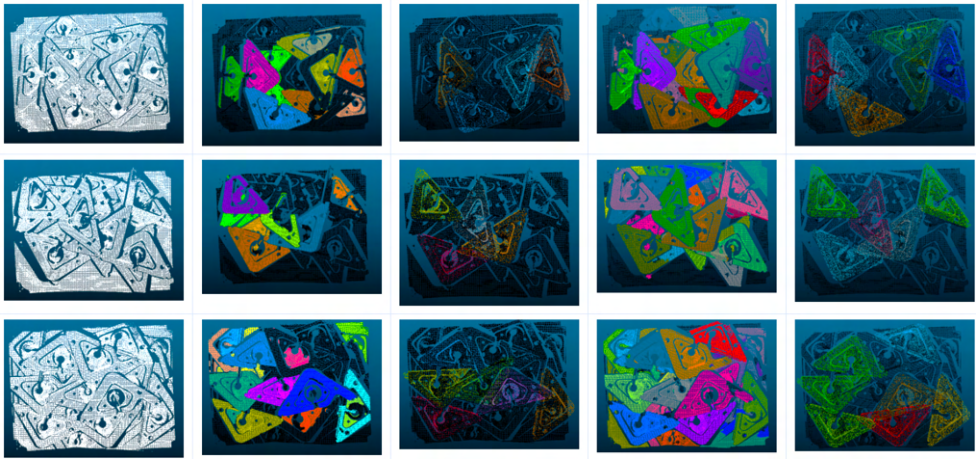


Fig. 7. Comparison between SPFormer [26] and proposed method. First column shows raw data. Second and fourth columns show instance segmentation results from SPFormer and our method. Third and fifth columns represent pose registration results from the instance segmentation results.

TABLE III
POSE ESTIMATION ERROR IN SIMULATION TEST DATA.

	position (cm)	rotation (deg)
Inference	0.182	1.243
Inference + ICP	0.074	0.178

also improve performance through multiple groups. This can be seen in Fig. 6.

2) performance on real data: Real-world test data was obtained with the Photoneo PhoXi S 3D sensor. The objects were placed in various positions and used as test dataset. We tested 50 scenes and reliable results were obtained even in environments with many occlusions. The segmentation results of the proposed method and SPFormer [26] were qualitatively compared, and part of them are shown in Fig. 7. From the last row of Fig. 7, it can be seen that the proposed method recognizes more objects and estimates poses.

We tested our pose estimation system on the generated test data including 90 scenes with 980 instances. The median value of the pose error is shown in Table. III. The translation error is 0.78 mm, and the rotation error is 0.178 degrees.

C. Robot Bin Picking

We deployed our network for the purpose of automating the retrieval of plastic components from bins containing real electronic products using robots. To ensure effective object handling, we established seven predetermined picking locations on the target object, evaluated by the following picking policy: 1) There are no obstructions in the way of the picking point's normal direction, and 2) The orientation of the picking point exists within a 50-degree range along the z-axis relative to the robot's base. From these eligible points, we selected the highest z-direction point as the final picking point. Our robot bin picking experiments were conducted using a KUKA robot model SR10R1420, which was equipped with a Zivid Two 3D sensor and a vacuum gripper, as depicted in Fig. 8. The estimated object instances and the chosen picking point are also visualized in Fig. 8. We performed the bin picking task 50 times, and our sim-to-real approach showed 88% pick success rate. Whereas PPF algorithm [16] predicted an average of 4.5 instances, our proposed algorithm predicted an average of 8.25 instances, indicating a significant 1.83-fold improvement in recall performance. You can view the actual experiment videos in the accompanying video material.

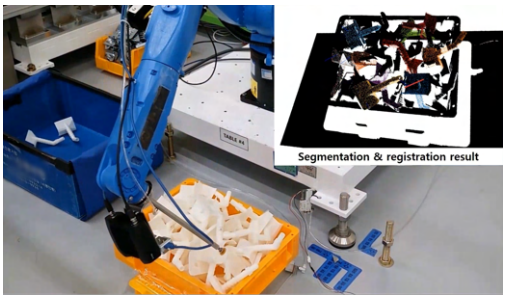


Fig. 8. Robot bin picking experiment using proposed framework.

V. CONCLUSION

In this paper, we present an innovative sim-to-real framework for estimating the poses of multiple objects in random bin picking scenarios. To address this challenge, we proposed a two-stage method, comprising 3D instance segmentation and registration steps, which collectively solve the problem. Our networks have solely undergone training using a synthetic dataset that faithfully replicates real-world bin picking scenarios by utilizing a physics simulator. The proposed approach enables our method to successfully recognize objects, even when only a part of an object is visible in obstructed scenes. Through a series of experiments, we demonstrate that our approach achieves a recall rate exceeding that of existing methods by more than 35% on standard benchmarks, all while maintaining comparable or even faster processing speeds. Furthermore, we demonstrated the seamless integration of our approach with real-world random bin picking tasks, leading to improved success rates.

REFERENCES

[1] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing mvtec itodd-a dataset for 3d object recognition in industry," in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 2200–2208.

[2] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6d object pose estimation dataset for industrial bin-picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2573–2578.

[3] X. Li, R. Cao, Y. Feng, K. Chen, B. Yang, C.-W. Fu, Y. Li, Q. Dou, Y.-H. Liu, and P.-A. Heng, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3961–3968, 2022.

[4] K. Chen, R. Cao, S. James, Y. Li, Y.-H. Liu, P. Abbeel, and Q. Dou, "Sim-to-real 6d object pose estimation via iterative self-training for robotic bin picking," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, pp. 533–550.

[5] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *2011 international conference on computer vision*. IEEE, 2011, pp. 858–865.

[6] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, pp. 876–888, 2011.

[7] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The international journal of robotics research*, vol. 30, no. 10, pp. 1284–1306, 2011.

[8] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3343–3352.

[9] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 540–14 549.

[10] 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 *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[11] 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 *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3828–3836.

[12] K. Kleeberger and M. F. Huber, "Single shot 6d object pose estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6239–6245.

[13] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[14] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[15] T. Rabbani and F. Van Den Heuvel, "Efficient hough transform for automatic detection of cylinders in point clouds," *Isprs Wg Iii/3, Iii/4*, vol. 3, pp. 60–65, 2005.

[16] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*. Ieee, 2010, pp. 998–1005.

[17] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[18] Y. Xu, S. Arai, D. Liu, F. Lin, and K. Kosuge, "Fpcc: Fast point cloud clustering-based instance segmentation for industrial bin-picking," *Neurocomputing*, vol. 494, pp. 255–268, 2022.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[20] Z. Dong, S. Liu, T. Zhou, H. Cheng, L. Zeng, X. Yu, and H. Liu, "Ppr-net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1773–1780.

[21] G. Gao, M. Lauri, J. Zhang, and S. Frintrop, "Occlusion resistant object rotation regression from point cloud segments," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[22] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," *arXiv preprint arXiv:1702.04405*, 2017.

[23] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," *arXiv preprint arXiv:1612.03777*, 2016.

[24] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[25] J. Fritsch, T. Kuhnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," *In Proceedings of the IEEE intelligent vehicles symposium (IV)*, 2013.

[26] J. Sun, C. Qing, J. Tan, and X. Xu, "Superpoint transformer for 3d scene instance segmentation," *arXiv preprint arXiv:2211.15766*, 2022.

[27] Z. J. Yew and G. H. Lee, "Regtr: End-to-end point cloud correspondences with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6677–6686.

[28] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3d instance segmentation on point clouds," *Advances in neural information processing systems*, vol. 32, 2019.

[29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[31] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9031–9040.

[32] L. Zeng, W. J. Lv, Z. K. Dong, and Y. J. Liu, "Ppr-net++: accurate 6-d pose estimation in stacked scenarios," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3139–3151, 2021.

[33] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2569–2578.

[34] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "Pointgroup: Dual-set point grouping for 3d instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, 2020, pp. 4867–4876.

[35] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia, "Instance segmentation in 3d scenes using semantic superpoint tree networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2783–2792.

[36] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[37] A. G. Buch, D. Kraft, S. Robotics, and D. Odense, "Local point pair feature histogram for accurate 3d matching." in *BMVC*, 2018, p. 143.

[38] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[39] S. Maneewongvatana and D. M. Mount, "It's okay to be skinny, if your friends are fat," in *Center for geometric computing 4th annual workshop on computational geometry*, vol. 2. Citeseer, 1999, pp. 1–8.

[40] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4096–4105.

[41] Y. Xu, S. Arai, F. Tokuda, and K. Kosuge, "A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color," *IEEE Access*, vol. 8, pp. 70 262–70 269, 2020.
[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.