International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:15, No:1, 2021

Deep Learning Based 6D Pose Estimation for Bin-Picking Using 3D Point Clouds

Hesheng Wang, Haoyu Wang, Chungang Zhuang

***Abstract*—**Estimating the 6D pose of objects is a core step for robot bin-picking tasks. The problem is that various objects are usually randomly stacked with heavy occlusion in real applications. In this work, we propose a method to regress 6D poses by predicting three points for each object in the 3D point cloud through deep learning. To solve the ambiguity of symmetric pose, we propose a labeling method to help the network converge better. Based on the predicted pose, an iterative method is employed for pose optimization. In real-world experiments, our method outperforms the classical approach in both precision and recall.

***Keywords*—**Pose estimation, deep learning, point cloud, bin-picking, 3D computer vision.

I. Introduction

BIN-picking is a crucial step in industrial automation. Since the demands of flexible manufacturing in industry 4.0, pose recognition by RGB/RGB-D images and point clouds has become a hot research field. In the real industrial scene, there is usually a pile of parts stacked randomly in a bin. The major problem is that the parts are from multiple categories and stacked with heavy occlusion. In this paper, we propose a method that can simultaneously detect mechanical parts, recognize their categories, and estimate their 6D poses from point clouds for the real bin-picking tasks.

Classical methods first extract features from scenes and treat pose estimation as a feature matching problem [1]. However, the handcrafted features are sensitive to the occlusion in bin-picking tasks. Therefore, recent works [6]-[9] use deep learning and deploy end-to-end solutions by considering the problem as a combination of object detection and pose estimation. These works usually use color and texture information from RGB/ RGB-D images, but mechanical parts often do not have these features. Dong et al. [9] proposed PPR-Net for point cloud inputs and achieved great results on Siléane Dataset [10] and take real-world experiments on the same objects of the dataset. But it was not applied on actual mechanical parts.

In this work, we propose a deep network that performs object detection and coarse pose registration simultaneously and refines the pose by Iterative Closest Point (ICP) [2]. The intuition of the method is simple: if our network has the ability to predict (or vote) a 3D location (like the object center and bounding-box corners) aligned on the same object by point-wise feature, we can get rough poses by predicting three locations of an object and matching them with the model in the same category. Then we could project the model box according to the coarse pose and select local points by the box. Finally, we sample the model points by occlusion-aware sampling and deploy ICP between the sampled model and the selected points to get precise 6D poses.

Chungang Zhuang is with the Shanghai Jiao Tong University, China (e-mail: cgzhuang@sjtu.edu.cn).

(a)

(b)

Fig. 1 Pose estimation using our method: (a) Real bin-picking scene and its corresponding point cloud. (b) The coarse pose registration (left) and final pose prediction (right)

In summary, our main contributions are as follows: (1) We propose a labeling method to regress the poses of symmetric objects through learning. (2) We propose a point-cloud based neural network which can detect objects and output the information of 6D poses at the same time. (3) We employ a clustering algorithm to predict poses and propose an iterative method based on ICP.

II. Previous Work

Classical 6D pose estimation approaches usually take both model and scene point clouds as input and output a transform based on the matching of handcrafted features. Papazov et al. [3] use an efficient RANSAC-like sampling strategy for robust pose registration. Drost et al. [1] proposed Point Pair Feature (PPF) based method to regress poses by descriptor voting scheme. PPF has been widely used since then. Abbeloos et al. [4] and Vidal et al. [5] further improved it for better performance. These methods have achieved success in special bin-picking tasks, but their performance will decrease due to occlusion and the variety of objects. Learning-based methods developed rapidly in recent years. Xiang et al. [6] used a convolutional network to estimate poses through RGB images.

Wang et al. [7] estimated 6D poses by an architecture (DenseFusion) which fuses the features of RGB images and corresponding point clouds. He et al. [8] achieved SOTA results through keypoints voting based on DenseFusion. These works gained great performance on the open dataset, but these approaches use RGB information and they were not designed for industrial scenes. Dong et al. [9] designed PPR-Net for industrial scenes with only 3D point clouds, which took experiments on both open dataset [10] and real tasks.

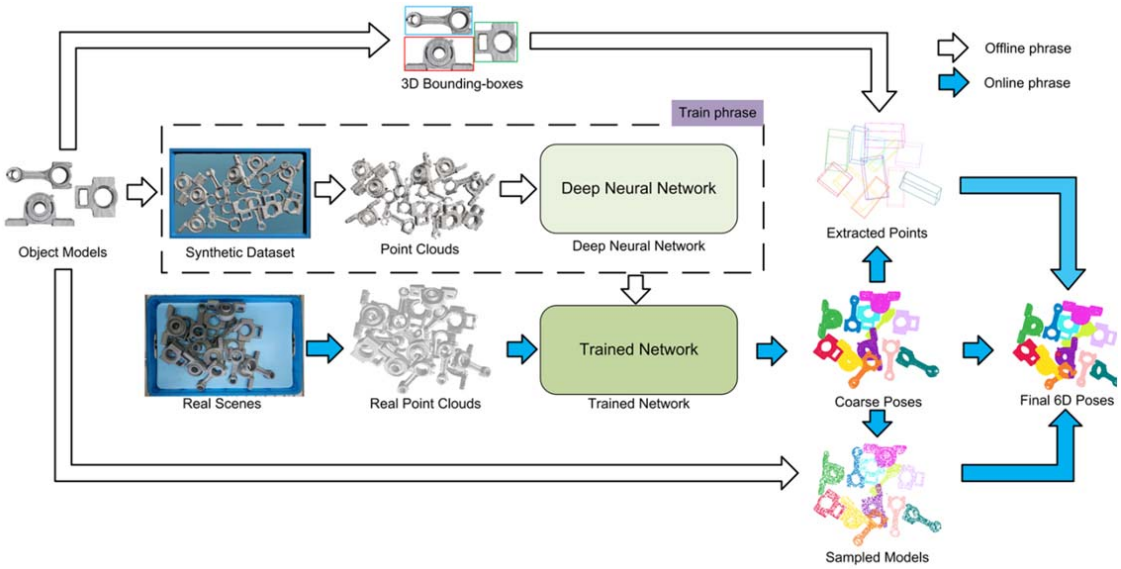


Fig. 2 Pipeline of our method for 6D pose estimation

Our work is based on the approaches of [12], [13], which predict 3D locations by learning-based Hough Voting. We extend the idea from instance segmentation to pose estimation. The proposed method takes experiments on real-world scenes and outperforms the classical approach which has been widely applied.

III. The Proposed Method

In this paper, we propose a general framework for detecting objects and estimating 6D poses on industrial bin-picking applications. This framework takes 3D point cloud as input and output 6D poses of workpieces.

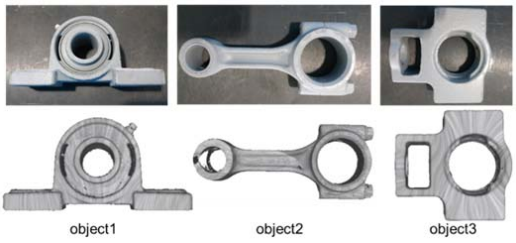


Fig. 3 Objects and corresponding models

For bin-picking tasks in industry, we usually have models of target objects. Hence, we prepare the virtual dataset by simulating stacks of these models. After annotation of the dataset, we introduce an architecture to predict coarse pose through each point votes three points on the object to which it belongs and grouping the votes. According to the coarse pose, we sample the model to make it more similar to the target point cloud in the scene which improves the accuracy of the final ICP. The pipeline of our method is shown in Fig. 2.

Section III *A* shows the process of generating and annotating the virtual dataset. Section III *B* introduces the predicting network. Section III *C* presents details of the 6D pose refinement after network outputs.

A. Pose Annotation on Symmetric Objects

The procedure of creating the virtual dataset is similar to [10]. The 3D models were first reconstructed by 3D scanning from multiple views. Fig. 3 shows the metal parts and corresponding models. Then the synthetic dataset could be obtained by simulation with these models. The domain randomization on simulation [11] has proven its capacity on sim-to-real transfer. Hence, we also considered these aspects like using random numbers of objects.

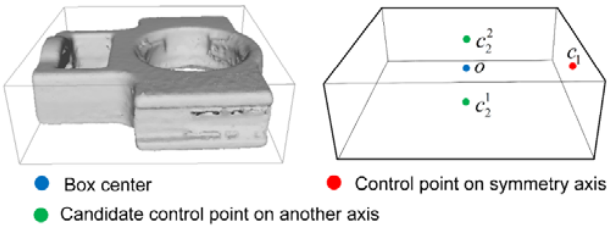


Fig. 4 The annotation method for symmetrical objects

As Fig. 3 shows, these objects all have symmetric properties on at least two surfaces, which means direct labeling of rotation often confuses the predicting network and does not gain good training results. To solve this problem, we propose an annotation method considering symmetry and ambiguous poses.

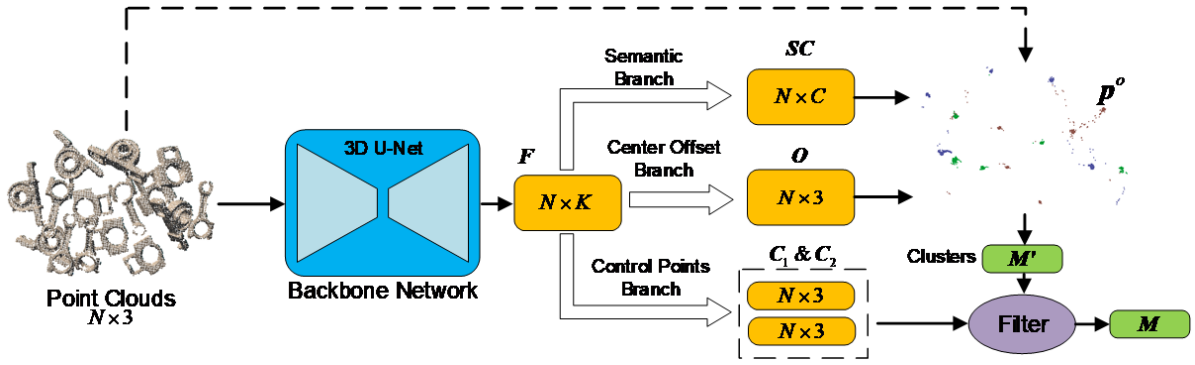


Fig. 5 The architecture of our neural network

The core idea of this method is to mark multiple ambiguous positions of the same object at the same time, then keep only one annotation according to certain rules, and finally, the network understands these rules through learning. As shown in Fig. 4, we mark three points for each object, represents the center transform, and the control points (c_1 and c_2^1 or c_2^2) together with o constitute a local coordinate system to represent the rotation. For ambiguity points c_2^1 and c_2^2, we first calculate their distance:

$$(\Delta x, \Delta y, \Delta z)^T = c_2^2 - c_2^1 \tag{1}$$

Then the main axis is determined by $argmax\{|\Delta x|, |\Delta y|, |\Delta z|\}$. The point with a smaller coordinate value on the main axis will be reserved as c_2.

B. Object Detection Network

In Fig. 5, we first voxelized the point clouds and feed them into the feature extraction network. We employ 3D U-net [14], a deep network which has the impressive ability to extract multi-scale feature information from 3D voxels as our backbone. Then it obtains the point-wise features $\boldsymbol{F}$ of size $N \times K$ by recover points from voxels. After that, we construct three branches with shared multi-layer perceptron (MLP) to consume $\boldsymbol{F}$ and predict semantic label S_i, center offset O_i, control points offsets C_{1i} and C_{2i} for each point p_i.

1) Semantic Segmentation Branch

As there are multiple types of objects in the same scene, it is necessary to classify the category S_i for each point. We feed features $\boldsymbol{F}$ through an MLP and produce semantic scores $\boldsymbol{SC}$ of size $N \times C$. $SC_{i,j}$ indicates the probability that p_i belongs to class j and S_i is obtained by (2):

$$S_i = argmax\{SC_{i,j}\}, j = 1, 2, ..., C \tag{2}$$

The semantic loss L_{sem} is the cross-entropy loss between $\boldsymbol{SC}$ and ground truth labels.

2) Center Offset Branch

The center offset branch is similar to [13]. It consumes $\boldsymbol{F}$ as input and predicts offset vectors $\boldsymbol{O}$ of size $N \times 3$, where O_i represents the spatial vector from p_i to its object center. The loss function L_{o_reg} and L_{o_dir} are the same with VoteNet [12] and PointGroup [13]:

$$L_{o\_reg} = \frac{1}{\sum_i m_i} \sum_i \left\| O_i - (o_i - p_i) \right\| \cdot m_i \tag{3}$$

$$L_{o\_dir} = -\frac{1}{\sum_i m_i} \sum_i \frac{O_i}{\|O_i\|_2} \cdot \frac{o_i - p_i}{\|o_i - p_i\|_2} \cdot m_i \tag{4}$$

$\boldsymbol{m}$ is a binary mask. $m_i = 1$ denotes p_i belongs to an object and $m_i = 0$ otherwise. o_i is the ground truth of center point and it has been obtained at III-A. L_{o_reg} is the L_1 loss of vectors. L_{o_dir} is irrelevant to the vector norm and would punish the network for biases of vector directions. The loss of this branch is defined as:

$$L_o = L_{o\_reg} + L_{o\_dir} \tag{5}$$

3) Control Points Offset Branch

This part is similar to the center offset branch because they are both solving the problem of predicting positions. The difference is that this branch takes two MLP to predict control points respectively. The evaluation function is defined as:

$$L_c = L_{c1\_reg} + L_{c1\_dir} + L_{c2\_reg} + L_{c2\_dir} \tag{6}$$

L_{c1_reg}, L_{c2_reg}, L_{c1_dir}, L_{c2_dir} are similar to (3), (4) while O_i

is replaced by C_{1i} or C_{2i} and o_i is replaced by c_{1i} or c_{2i} .

C. Pose Optimization

After the deep network, we obtained semantic labels $\boldsymbol{S}$, offset vectors $\boldsymbol{O}$, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. As Fig. 5 shows, we will first detect objects by grouping points. Then we regress coarse poses from $\boldsymbol{O}$, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ for each cluster. Hence, we sample the model according to coarse poses and refine poses by ICP.

1) Point Clustering

Towards the point clouds which have semantic labels, the clustering algorithm deployed in PointGroup [13] has achieved impressive performance on shifted points. We employ $\boldsymbol{O}$, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as our shift vectors:

$$p_i^o = p_i + O_i, p_i^{c1} = p_i + C_{1i}, p_i^{c2} = p_i + C_{2i} \quad (7)$$

Then the clustering algorithm is performed on $\boldsymbol{p^o}$ because centers of different objects often have a distance for their collision shapes, which makes clustering easier. After the clustering, we obtained clusters $\boldsymbol{M'}$. In order to obtain more concentrated clustering results for pose regression, we filter each cluster by ball-query filtering on $\boldsymbol{p}^{c1}$ and $\boldsymbol{p}^{c2}$ to get final clusters $\boldsymbol{M}$ and M_k denotes the k_{th} cluster.

2) Coarse Pose Regression

The center and control points prediction of the cluster M_k can be estimated as follows:

$$o_k^{pred} = \frac{\sum_i p_i^o}{N_k}, c_{1k}^{pred} = \frac{\sum_i p_i^{c1}}{N_k}, c_{2k}^{pred} = \frac{\sum_i p_i^{c2}}{N_k}, p_i \in M_k \quad (8)$$

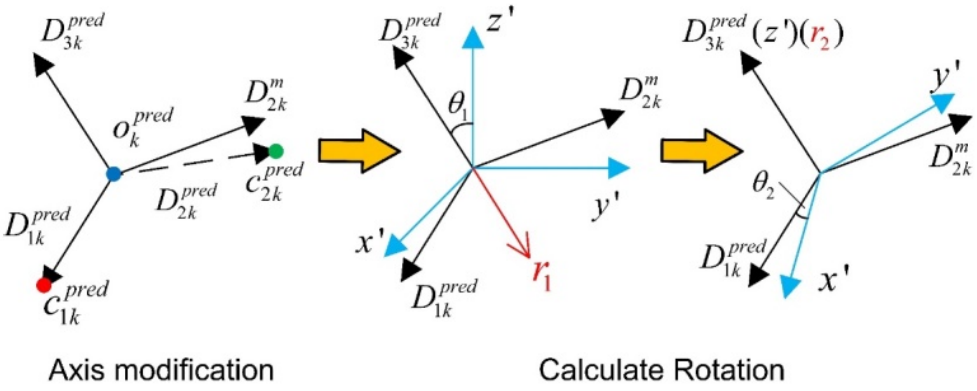


Fig. 6 The procedure to regress rotation

As Fig. 6 shows, o_k^{pred} is the origin of the local coordinate system, and $D_{1k} = c_{1k}^{pred} - o_k^{pred}$ and $D_{2k} = c_{2k}^{pred} - o_k^{pred}$ represent the axis direction vectors. The third axis is calculated by $D_{3k} = D_{1k}^{pred} \times D_{2k}^{pred}$. Then we use $D_{2k}^m = D_{3k}^{pred} \times D_{1k}^{pred}$ to modify the axis direction and ensure orthogonality of coordinate axes. Then the rotation matrix $\boldsymbol{R}_k$ can be calculated by Rodrigues' rotation formula [15] and rotation vectors (r_1 and r_2) in Fig. 6. Hence, the transform matrix of the coarse pose is obtained:

$$\boldsymbol{T}_k^0 = [\boldsymbol{R}_k, o_k^{pred}] \quad (9)$$

3) Pose Refinement

ICP is a general method in point clouds pose optimization. However, its performance will decrease due to the heavy occlusion in bin-picking tasks. Furthermore, it is sensitive to initial pose selection. To overcome its limitations, we propose an iterative method based on ICP in Fig. 7. $\boldsymbol{T}_k^i$ is the transform matrix after i_{th} iteration and the initial matrix is $\boldsymbol{T}_k^0$. Before ICP, we sample the model point cloud according to the occlusion of $\boldsymbol{T}_k^i$ .We also select scene points by projecting model bounding-box through $\boldsymbol{T}_k^i$. Then the sampled model, selected points and $\boldsymbol{T}_k^i$ are fed into ICP to get $\boldsymbol{T}_k^{i+1}$.

IV. Experiments

In this section, we evaluate our method on real-world scenes captured by an industrial 3D scanner. To prove its capacity on pose estimation, we also compared its mean precision and mean recall with the classical approach: PPF with ICP.

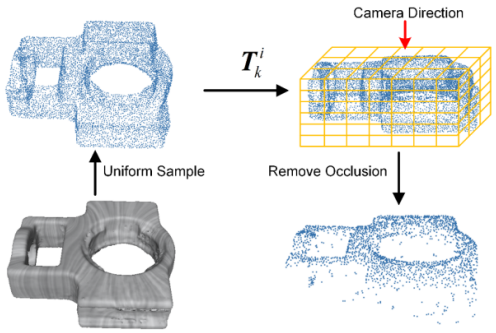
A. Real World Data

Fig. 8 (a) shows the hardware for experiments. A 3D scanner produced by Photoneo is using as our point clouds producer. The perspective of the sensor and its output point cloud is shown in Fig. 8 (b). We prepared 50 different scenes for the experiment. Each scene has 14-17 objects from three categories.

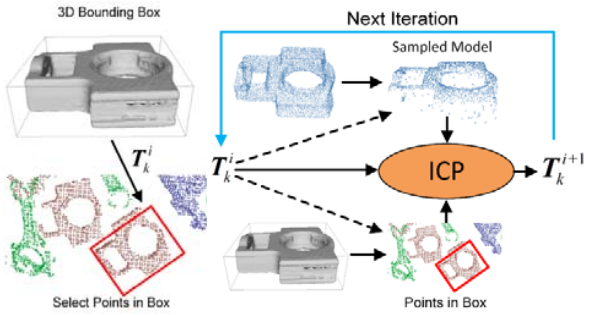
B. Implementation Details

Before being fed into the network, we sample each scene by voxel sampling and the voxel size is 0.003 m. The synthetic dataset contains 1530 scenes for training. The voxel size using in occlusion sampling is 0.002 m. The mean distance threshold is 0.006 m, which determines if a prediction is considered as

positive after the final ICP. We implement PPF with Halcon for comparison.



(a)



(b) (c)

Fig. 7 The pose refinement: (a) The sample method with occlusion. (b) Scene points selection. (c) The iterative method based on ICP

(a) (b)

Fig. 8 (a) Experiment setting. (b) Photo and the corresponding point cloud captured by 3D scanner

C. Results and Discussion

The method is employed on 50 real-world scenes. Several predictions are visualized in Fig. 9. We also implement PPF in the same scenes. The mean precision and mean recall of all the scenes are shown in Fig. 10. The blue bars are the results of our method and the red bars are the results of PPF. As Fig. 10 shows, the proposed method achieves higher mean precision and mean recall than PPF with ICP does. Our method outperforms PPF by 16% in precision and 6% in recall.

(a) (b) (c)

Fig. 9 Real scenes and corresponding predictions (a) Scene photos. (b) Coarse poses produced by the network. (c) Final poses refined by our method

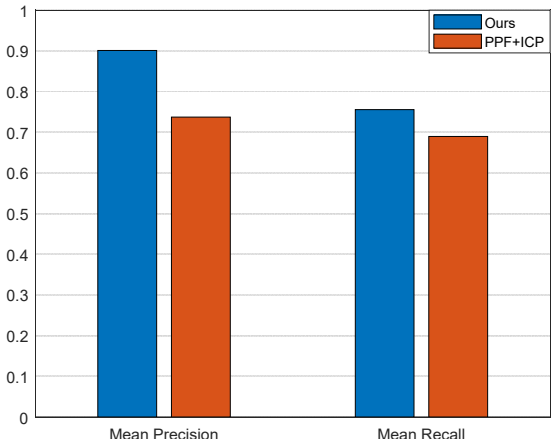


Fig. 10 Comparison of our method and PPF

V. Conclusions

In this paper, based on the intuition that the coarse poses of objects could be regressed by three 3D locations, we proposed a labeling method for symmetry objects. Hence, we employed a network, which takes point clouds as input and trained by these labels, to vote centers and control points. With the output of the network, a clustering method is adopted to detect objects and get their coarse poses. Then the coarse poses were refined with an iterative method based on ICP. In the experiments, we take real-world scenes for tests. Our method outperformed PPF in both precision and recall.

Acknowledgment

This research work is supported by NSFC-Shenzhen Robot Basic Research Center project (U2013204) as well as the National Natural Science Foundation of China under Grant No.51775344.

References

[1] 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, 2010, pp. 998–1005.
[2] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.
[3] C. Papazov and D. Burschka, "An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes," in Computer Vision-ACCV 2010, PT I, Heidelberger Platz 3, D-14197 Berlin, Germany, 2011, vol. 6492, no. I, pp. 135–148.
[4] W. Abbeloos and T. Goedemé, "Point Pair Feature Based Object Detection for Random Bin Picking," in 2016 13th Conference on Computer and Robot Vision (CRV), 2016, pp. 432–439.
[5] Vidal, C. Lin, and R. Martí, "6D pose estimation using an improved method based on point pair features," in 2018 4th International Conference on Control, Automation and Robotics (ICCAR), 2018, pp. 405–409.
[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes." 2018.
[7] C. Wang et al., "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3338–3347.
[8] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11629–11638.
[9] Z. Dong et al., "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), 2019, pp. 1773–1780.
[10] 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), 2019, pp. 2573–2578.
[11] Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23–30.
[12] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough Voting for 3D Object Detection in Point Clouds," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9276–9285.
[13] L. Jiang, H. Zhao, S. Shi, S. Liu, C. -W. Fu, and J. Jia, "PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4866–4875.
[14] O. Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016. 19th International Conference. Proceedings: LNCS 9901, 2016, pp. 424–32.
[15] K. K. Liang, "Efficient conversion from rotating matrix to rotation axis and angle by extending Rodrigues' formula." 2018.