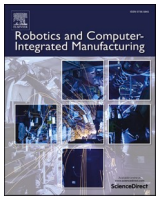
Robotics and Computer–Integrated Manufacturing 86 (2024) 102671

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



AttentionVote: A coarse-to-fine voting network of anchor-free 6D pose estimation on point cloud for robotic bin-picking application

Chungang Zhuang *, Haoyu Wang, Han Ding

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, PR China

ARTICLE INFO

Keywords:
Pose estimation
Voting network
Point cloud
Industrial parts
Robotic bin-picking

ABSTRACT

Current state-of-the-art pose estimation methods are almost launched on segmented RGB-D images. However, these methods may not apply to more general industrial parts due to a lack of texture information and high-occlusion of stacked objects. This article establishes an end-to-end pipeline to synchronously regress all potential object poses from an unsegmented point cloud. The point pair features (PPFs) are first extracted and then fed into a PointNet-like backbone for obtaining the point-wise features. Based on the center voting, a coarse-to-fine voting architecture is proposed to extract instance features instead of implementing instance segmentation. A lightweight three-dimensional (3D) heatmap is leveraged to cluster votes and generate center seeds. Further, an attention voting module is constructed to fuse point-wise features into instance-wise features adaptively. Ultimately, the suggested network regresses object poses with a quaternion loss to handle the symmetric puzzle. The network holds the advantage of producing the final pose prediction without any post-processing steps like non-maximum suppression (NMS) or any pose refinement modules like iterative closest point (ICP). The proposed network is evaluated on the public Fraunhofer IPA dataset, which demonstrates the robustness of the pose estimation network with much better performance. Meanwhile, the network is further validated on our synthetic and real-world datasets of industrial parts for robotic bin-picking.

1. Introduction

Bin-picking of randomly arranged objects is a very tedious and slow task in the field of conventional industrial automation, and the primary goal of robotic manipulation is to automate the manufacturing process entirely and perform the repetitive tasks as much as possible [1–4]. In the application of robotics, performing bin-picking is still a challenging issue, in which accurate six-dimensional (6D) pose estimation is one of the key technologies to be solved for achieving effective bin-piking manipulation [5–7]. The objective of pose estimation is to estimate the six degrees of freedom of rigid transformation (i.e., translation and rotation) from the object coordinate system to the camera reference frame. Many deeper investigations have already been conducted to tackle this problem and achieved promising results [8,9]. However, limited by available data acquisition technique, the object pose recovery is still a difficult task due to sensor noise, highly occluded objects, and cluttered scenes [10].

In recent works, the data-driven methods have shown absolute predominance in pose estimation for their efficiency and robustness compared to the traditional methods. Inspired by the excellent performance of convolutional neural networks (CNNs) on two-dimensional (2D) object detection, several research works were completed by CNN for pose estimation based on RGB images [11,12]. However, the incomplete geometric information limits the performance of the RGB-based methods, especially for challenging scenarios such as poor lighting conditions, textureless objects, and cluttered scenes. Thus, the RGB-D based methods emerge and become popular with the advent of depth sensors in pose estimation. Some methods [9,13] have demonstrated state-of-the-art performance on RGB-D public datasets such as YCB-Video [12] and LINEMOD [14]. These methods generally follow the pipeline of performing the semantic segmentation on the RGB images firstly, and then cropping the RGB and depth images with the predicted bounding box into per instance region [15–20]. This strategy helps to filter the irrelevant information and learn the normalized input, but increases the number of the batch and propagation. Besides, many depth-only methods also rely on semantic segmentation [21,22].

This article focuses on the pose estimation problem for the vision-guided robotic bin-picking application. Bin-picking is a typical task in

* Corresponding author.
E-mail address: cgzhuang@sjtu.edu.cn (C. Zhuang).

https://doi.org/10.1016/j.rcim.2023.102671
Received 12 March 2023; Received in revised form 3 August 2023; Accepted 4 October 2023
Available online 7 October 2023

the robotic manipulation field and has a wide range of industrial applications in manufacturing and logistics factories. In this work, different categories of objects (such as industrial parts) are randomly stacked into a high occlusion status, and an industrial robot equipped with visual sensor is required to perceive the objects and picks them up sequentially from the occluded scene. For the vision module, this work pursues simultaneous pose predictions of multiple objects, and then an optimal one is selected for the grasping manipulation. Therefore, the pre-processing step of instance segmentation is redundant and not necessary. In this way, the point cloud-based pose estimation method, which can directly process and analyze the entire point cloud including all objects of the whole scene, may be more appropriate than the segmentation-based methods as mentioned above.

In this work, an end-to-end point cloud-based deep learning approach is suggested to solve the 6D pose estimation problem for the industrial bin-picking application. The work of this article aims to take the 3D point cloud of the whole scene as input and directly outputs the 6D pose of each instance. First, in place of semantic segmentation, a heatmap for analyzing object localization is constructed in 3D space. Then, based on the idea of the Hough voting in 3D object detection [23], the instance proposals are directly generated while extracting features from point cloud without extra anchor hypotheses. Finally, combined with the attention mechanism, an agglomerative method is suggested for the instance-wise pose regression, which is defined as the attention voting module. In this way, the instance features are extracted in terms of votes as a normalized expression and thus can be fed as a batch into the shared layer. This avoids the processing of the traversal and multiple propagations when handling all objects in the scene at the same time.

In summary, an end-to-end pose estimation network is established with only point cloud as input. The major contributions of this work are presented as follows:

(1) A new proposal module with heatmap, which simultaneously predicts multiple poses over the entire point cloud and effectively avoids repeated prediction or omission according to the experimental results, is designed for the anchor-free pose estimation.
(2) An attention voting module is constructed to aggregate point-wise feature into instance features. This module can effectively improve the accuracy of the network regression compared with the general pooling methods and avoid extra refinement post-processing.
(3) The angle increment is directly computed by quaternion for regression instead of those based on the average distance of model points or key points. Thus, the network can effectively deal with the symmetric puzzle by comparing the errors from the multiple symmetric ground-truth poses.

The rest of this article is organized as follows. Section 2 briefly reviews the related methods of pose estimation based on point clouds. Section 3 presents the pipeline of the proposed coarse-to-fine voting network for the pose estimation on point cloud. The performance evaluation of the pose prediction network and the robotic bin-picking experiments are conducted in Section 4. Finally, some concluding remarks on this study are summarized in Section 5.

2. Related work

2.1. Pose estimation using point cloud

Due to its unstructured and unordered nature, it is infeasible to feed raw point cloud into network. In early work, point cloud was usually converted into voxels [24] or only used for perspective-n-point algorithm [25] or ICP refinement [12,26,27], which failed to utilize the rich geometric information of point cloud. PointNet [28] and PointNet++ [29], which were well-known as the backbone of point cloud, made it possible to extract features and estimate object poses from point cloud. DenseFusion [8] was proposed to combine the point cloud features from PointNet with the color features from CNN at the dense pixel level. Some works followed the idea of the feature fusion and achieved state-of-the-art performances [9,13]. CloudPose [30] was regarded as the first learning-based method for pose estimation only using point cloud. It employed two branches of PointNet for the translation and rotation regressions. CloudAAE [31] argued that the domain discrepancy between the synthetic and real-world images was considerably smaller and easier to be reduced for the depth information. It adopted an augmented autoencoder [32] to learn a potential object poses from the synthetic depth data. StablePose [33] suggested to learn the pose inference based on the geometrically stable patches extracted from point cloud. OVE6D [21] regressed viewpoint, in-plane rotation, and translation as the pose in a cascaded manner from a single depth image and a target object mask. It constructed a viewpoint codebook from 3D mesh models, and thus can predict the poses of the untrained objects as the corresponding meshes are given. It achieved state-of-the-art performances on the challenging T-LESS dataset [34].

2.2. PPF-based pose estimation

The PPF-based method [35] was a powerful and effective pose estimation method based on the Hough voting scheme. A four-dimensional feature named PPF was proposed and the geometric correspondence between PPFs of the model and scene point clouds was matched by using a fast voting scheme. Many investigations have been carried out to improve the performances of different stages of the PPF-based method including sampling [36–38], embedding [39], voting [39,40], and post-processing [37]. However, the PPF-based method is limited by efficiency of multiple loop iterations, sensitivity to noise, occlusion, and clutter. In recent works, the PPF was also combined into the learning-based investigations for its natural characteristics. PPFNet [41] fed the PPFs into PointNet [28] for point matching. The traditional PPF was extended and applied to the category-level PPF (CPPF) voting method [22] for predicting object centers, orientations and scales, and the invariant embeddings including CPPF information were fed into the network for the category-level pose estimation.

2.3. Center-based methods

The PPF-based method [35] essentially converted pose estimation into voting in a coded Hough space. Recently, many regression networks with point clouds directly performed the voting procedure in the 3D Euclidean space or in the 6D pose space. The idea of voting in the centroid space originated from 3D object detection. VoxelNet [24] augmented each point with a relative offset from the mean centroid of voxels. VoteNet [23] predicted the point-wise centers and performed voting in the centroid space to cluster points of an instance for object detection. PointGroup [42] implemented a similar voting strategy on the dense points for instance segmentation. AFDet [43] and CenterPoint [44] proposed a 2D center heatmap for anchor-free 3D object detection. PPR-Net [45] voted in the pose space and PPR-Net++ [46] voted in the centroid space for the instance segmentation as well as the pose estimation. In this article, a coarse-to-fine centroid voting approach is proposed for the anchor-free pose estimation, as shown in Fig. 1.

3. Method

The pipeline of the proposed pose estimation network is presented in Fig. 1. A sampled point cloud with normal information of the whole scene is used to regress the object center coordinates of each seed point. Then, the network generates the proposal centers of objects according to the heatmap information and fuses the point-wise features with the attention mechanism, ultimately makes the instance-wise predictions without any post-processing steps.

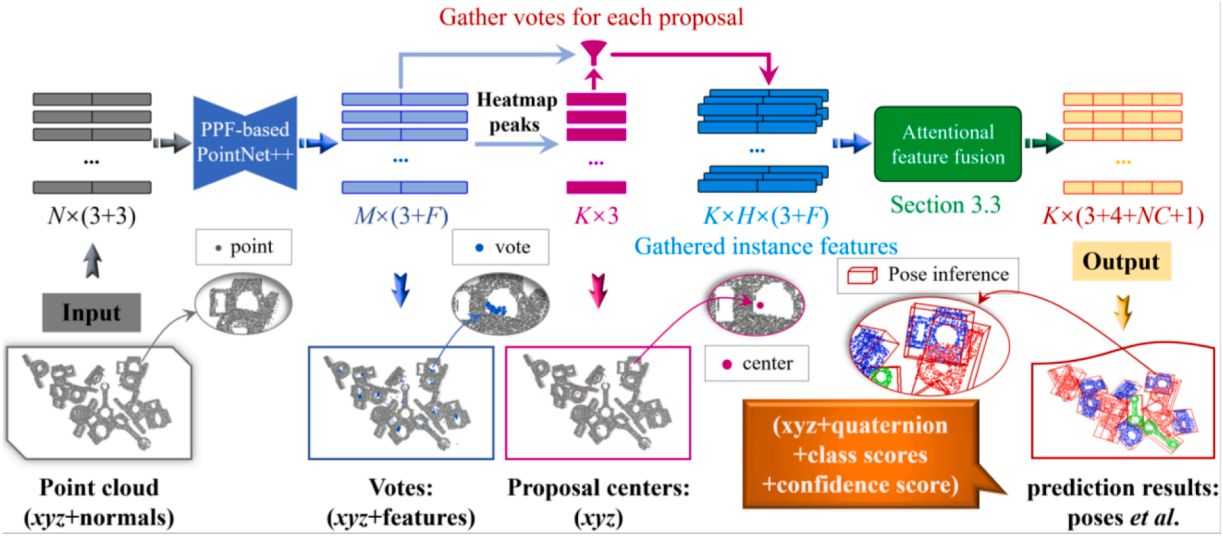


Fig. 1. Illustration of the AttentionVote network pipeline. The network learns the input point cloud with the *xyz* coordinates and the corresponding normals through the backbone of the PPF-based PointNet++, then generates votes with high-dimensional features. Several proposal centers are generated from the heatmap of votes. Each proposal gathers votes as the instance features. After the attentional feature fusion and the shared multilayer perceptron (MLP), the network directly outputs the final pose estimation results without any post-processing steps.

3.1. Learning of the rotation-invariant votes

Following the basic idea of the center voting, the center predictions as votes are generated from the input point cloud. This stage includes two steps: learning rotation-invariant features from point cloud through a backbone network and generating votes by learning the residual features through a voting module.

Backbone for point cloud. In order to extract the rotation-invariant features from local geometric information and global contextual relationships, PointNet++ [29] is adopted as the foundation of the backbone due to the point-wise equivalency of the shared MLP and the receptive field flexibility of the set abstraction (SA) layers, which is capable of extracting the local and global features in the unsegmented intensive-stack scene. Besides, the PPFs [35] are hand-crafted features extracted from point cloud and the normals are free of restriction by rotation and scale. Inspired by PPFNet [41], the PPFs are combined with the PointNet++ architecture as the backbone to augment the input data. In detail, for a seed point and its neighbor sampled points in a SA layer, their PPFs as the additional local features are computed and fed into the shared MLP. Overall, the backbone takes the point cloud of N points with the *xyz* coordinates and the normal vectors as input, and outputs a subset of seed points $\{s_i = [x_i; f_i^{seed}], i = 1, \ldots, M\}$, where $x_i \in \mathbb{R}^3$ is the *xyz* coordinates, $f_i^{seed} \in \mathbb{R}^F$ is the learned F-dimensional features, and M is the number of points.

Residual features for voting. A shared voting module [23] is leveraged to learn votes from the seed points. The module implemented with a standard shared MLP learns a $(3 + F)$-dimensional residual feature $[\Delta x_i; \Delta f_i^{seed}]$ from the F-dimensional feature f_i^{seed} of a seed point s_i, where $\Delta x_i \in \mathbb{R}^3$ is the centroid offset and $\Delta f_i^{seed} \in \mathbb{R}^F$ is the feature offset for regression. In this way, a subset of votes $\{v_i = [y_i; g_i], i = 1, \ldots, M\}$ with $y_i = x_i + \Delta x_i$ and $g_i = f_i^{seed} + \Delta f_i^{seed}$ are generated for the following proposal generating module in Section 3.2 and instance feature fusion module in Section 3.3. Here, y_i is supervised by a regression loss $L_{vote}^{(i)}$:

$$L_{vote}^{(i)} = \| y_i - \hat{y}_i \|, \tag{1}$$

where $\hat{y}_i$ is the ground-truth center. In order to improve the voting accuracy, the warm-up of the network model in terms of the loss function is needed in the early training phase.

3.2. Heatmap for anchor-free proposal generation

This module aims to generate K proposals from M predicted centers. Compared to the traditional Hough voting methods, VoteNet [23] adopted a simple method to execute the Farthest Point Sampling (FPS) in the Hough space for generating the proposals. Though adapting to the end-to-end pipeline, the FPS neglects the spatial distribution pattern of votes, and so the proposals with randomness may deviate from the ground-truth centers or overlap on the same one. As mentioned in Section 2.3, an architecture of 3D heatmap is leveraged for the anchor-free proposal generation, which inherits the idea of the traditional Hough voting to divide the Hough space into the voting bins. The centers of votes are voxelized in the 3D Euclidean space by an appropriate step related to the model scale, and each voxel counts the magnitude of votes inside it as heat intensity. Then a 3D Gaussian filter implemented with the 3D sparse convolution [47,48] is applied to generate a smooth heatmap, as presented in Fig. 2, where the heat intensity shows the probability of proposals. The higher a voxel heat intensity reaches, the smaller the deviation between the predicted and ground-truth centers is likely to be. In the implementation, the top K local peaks are taken into account and the 3D center coordinates $\{p_k, k = 1, \ldots, K\}$ of the corresponding voxels are regarded as the center seeds. To avoid the interferences of the noise and redundancy proposals, only the peaks, that reach a predefined threshold, are taken into consideration during the inference stage. By employing the spatial distribution analysis, the proposed network achieves NMS-free proposal prediction in practice.

3.3. Attention module for instance feature fusion

Vote clustering through uniform sampling. Deriving from a center seed, the neighbor votes are aggregated to fuse features for the instance regression. Given a proposal p_k, the votes located in the sphere search domain are gathered by a k-dimensional tree in the 3D Euclidean space and are taken as the candidates. The sphere search domain is defined as:

$$S_k = \left\{ v_j^{(k)} \,\middle|\, \| v_j^{(k)} - p_k \| \leq r, v_j^{(k)} \in V, j = 1, \ldots, H_k \right\}, \tag{2}$$

where r is the radius of the sphere, v_j is defined by $\{v_j = [y_j; g_j], j = 1, \ldots, M\}$, V is the set of votes, and H_k is the number of votes in S_k. In order to keep synchronism in the face of vote count difference among proposals,

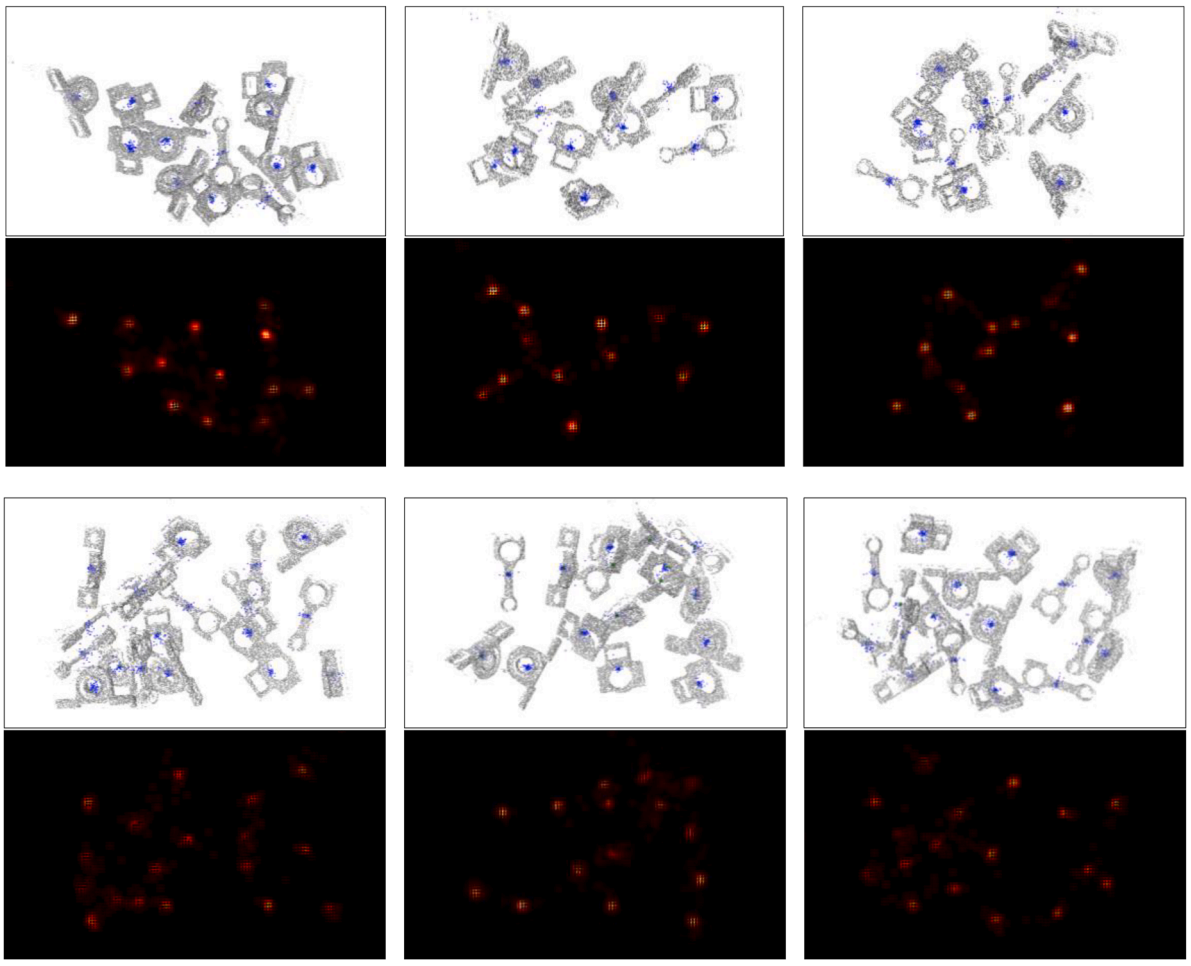


Fig. 2. Heatmaps from voting centers for proposals. In the first and third rows, the raw point clouds are marked in gray. The blue points are the predicted vote centers, and the green points are the ground-truth centers for reference. The second and fourth rows present the corresponding heatmaps of vote centers. The warmer a voxel color appears, the higher the probability of an object center located in the voxel is.

the H candidates are uniformly sampled for the proposal p_k as:

$$C_k = \left\{ v_s^{(k)} \middle| v_s^{(k)} \in S_k, s = 1, \ldots, H \right\}, \tag{3}$$

It should be noted that $v_s^{(k)}$ may be repeatable.

Deep Hough voting and feature fusion. Different from traditional Hough voting, deep Hough voting is expected to deal with votes with high-dimensional features discriminatively. Thus, as illustrated in Fig. 3 (a), a shared attention module is proposed to allocate the weights of the vote features self-adaptively. The attention voting module consists of two shared MLP branches, which separately learn the instance features to be fused and the attention weights of features. For a vote $v_s^{(k)} = [y_s^{(k)}; g_s^{(k)}]$, this module takes the locally normalized feature $[z_s^{(k)}; g_s^{(k)}]$ as input where $z_s^{(k)} = (y_s^{(k)} - p_k)/r$ and outputs an instance feature vector $f_s^{ins(k)}$ and an attention weight $w_s^{(k)}$ with the same dimension. Then, the proposal feature can be computed as:

$$\begin{aligned} f_k^{prop} &= \sum_{s=1}^{H} \mathrm{softmax}\left(w_s^{(k)}\right) f_s^{ins(k)} \\ &= \frac{\sum_{s=1}^{H} \exp\left(w_s^{(k)}\right) f_s^{ins(k)}}{\sum_{s=1}^{H} \exp\left(w_s^{(k)}\right)}. \end{aligned} \tag{4}$$

Fig. 3(b), which is the matrix version of the attention module, is proposed as a contrast. The attention weights are obtained from the matrix product of vote features and weight matrix (red color) which is set as the network parameter and trained directly. In experimental investigations, the above-mentioned modules are compared to verify their effectiveness. Finally, f_k^{prop} is fed into a head module of the shared MLP to obtain a regression result of p_k. The details of the heads and the loss functions are introduced in Section 3.4.

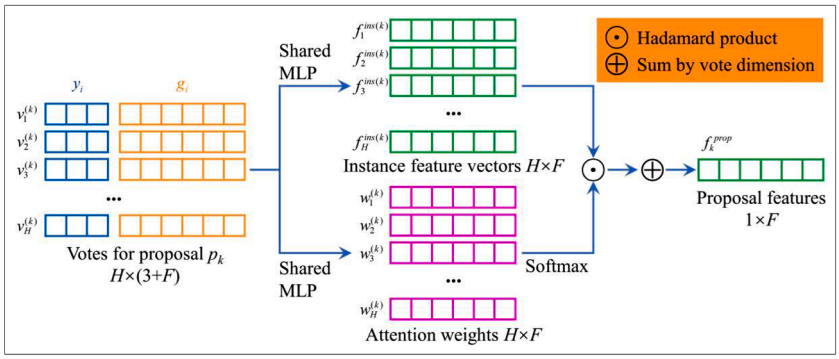
This adaptive fusion mechanism can reduce the interference from the irrelevant context information caused by the multilevel spherical receptive fields on the unsegmented point cloud. More details and performance comparisons are provided in Section 4.

3.4. Loss function definition

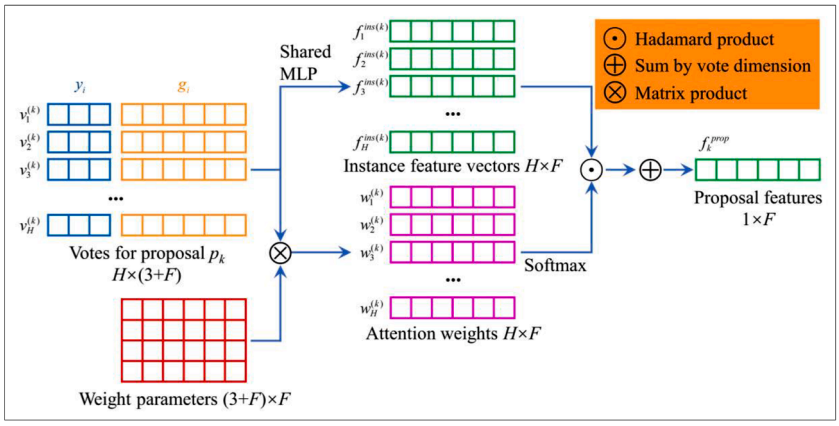
The proposed network outputs an embedded vector for a proposal that consists of center, rotation expressed by quaternion, semantic classification score, and confidence score. The loss function supervises $B \times K$ proposals in one backpropagation, where B is the batch size of the scenes and K is the number of proposals in one scene.

Assigning instance labels. As center proposals are generated from voting instead of semantic segmentation, the most possible ground-truth object in the scene is chosen as the training label corresponding to each proposal. Here, we explicitly assigned the instance label with the closest distance to a proposal center, and a threshold value is set to distinguish whether a proposal is valid or not, as shown in Fig. 4. Only the proposals, the distances of which from the label centers are within the predefined threshold, are activated and supervised.

Supervising an instance. The loss of instance is decoupled into center loss, rotation loss, semantic classification loss, and confidence

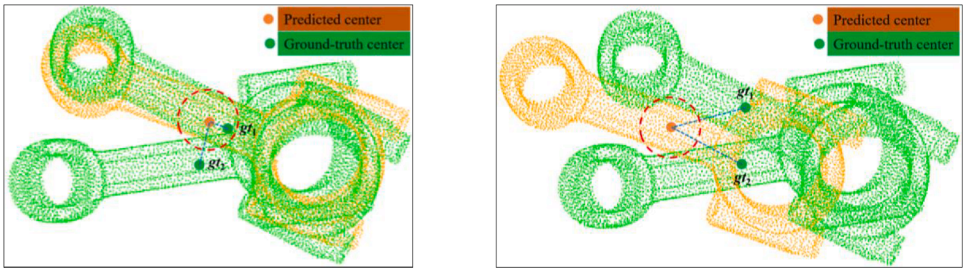


(a) The shared MLP version of the attention module.



(b) The matrix version of the attention module.

Fig. 3. The architecture of the attention module. (a) The shared MLP version of the attention module. The attention weights are obtained from a shared MLP branch. (b) The matrix version of the attention module is proposed as a contrast. The attention weights are obtained from the matrix product of vote features and weight matrix (red color) which is set as the network parameters and trained directly. (a) The shared MLP version of the attention module. (b) The matrix version of the attention module.



(a) Valid proposal gt_1. (b) Invalid proposal.

Fig. 4. Label assignment. The left figure (a) has one valid proposal gt_1 in terms of the predefined threshold value, and there are no valid proposals in the right figure (b). (a) Valid proposal gt_1. (b) Invalid proposal.

loss. The $L2$ loss $L_{cen}^{(k)}$ is adopted for the center regression, and the standard cross entropy loss $L_{sem}^{(k)}$ is used for the semantic classification. The rotation is directly regressed by integrating it into the end-to-end pipeline instead of those based on geometric models such as the average distance of model points [8] or key points [13]. The quaternion is employed to represent rotation due to its compact and nonsingular representation [49], and the linear interpolation for regression [50]. Given an embedded rotation $q \in \mathbb{R}^4$, it is a surjection from the group of unit quaternion $\overline{q} = q / \| q \|$ to the rotation group $SO(3)$. Given a predicted rotation $q_k = [w_k, x_k, y_k, z_k]$ with its unit quaternion $\overline{q}_k = \overline{w}_k + \overline{x}_k \mathrm{i} + \overline{y}_k \mathrm{j} + \overline{z}_k \mathrm{k}$ and the ground-truth quaternion $\widehat{q}_k = \widehat{w}_k + \widehat{x}_k \mathrm{i} + \widehat{y}_k \mathrm{j} + \widehat{z}_k \mathrm{k}$, the angular displacement can be computed by the orientation increment Δq from $\overline{q}_k$ to $\widehat{q}_k$ with the relationship $\widehat{q}_k = \overline{q}_k \otimes \Delta q$, where $\otimes$ denotes the quaternion product. The orientation increment Δq is stated as:

$$\begin{aligned} \Delta q &= \overline{q}_k^{-1} \otimes \widehat{q}_k \\ &= (\overline{w}_k - \overline{x}_k \mathrm{i} - \overline{y}_k \mathrm{j} - \overline{z}_k \mathrm{k}) \otimes (\widehat{w}_k + \widehat{x}_k \mathrm{i} + \widehat{y}_k \mathrm{j} + \widehat{z}_k \mathrm{k}) \\ &= \cos\frac{\phi}{2} + \sin\frac{\phi}{2} \mathrm{u}, \end{aligned} \tag{5}$$

where ϕ is the angular displacement rotated by quaternion Δq in 3D space and it is taken as the loss function:

$$\begin{aligned} L_{rot}^{(k)} &= \phi \\ &= 2\arccos(\overline{w}_k\widehat{w}_k + \overline{x}_k\widehat{x}_k + \overline{y}_k\widehat{y}_k + \overline{z}_k\widehat{z}_k) \\ &= 2\arccos(\overline{q}_k\cdot\widehat{q}_k), L_{rot}^{(k)} \in [0, 2\pi). \end{aligned} \tag{6}$$

Furthermore, an optional or explicit method is to constrain the regression with rotation angle $\phi \leq \pi$ [50] according to the shortest rotation distance in 3D space, namely:

$$L_{rot}^{(k)} = 2\arccos(|\overline{q}_k\cdot\widehat{q}_k|), L_{rot}^{(k)} \in [0, \pi). \tag{7}$$

However, the performance of Eq. (7) is a little worse than that of Eq. (6) with more details provided in Section 4.3.

For the cyclic symmetry with multiple ground truth poses, the L_{rot} function is computed for each pose and the corresponding minimum one is selected. For the revolution objects with infinite true poses, it degenerates into the angle between the symmetry axes of the predicted and ground-truth rotations. Take the $\boldsymbol{z}$-axis of symmetry as an example, the rotation loss can be obtained by:

$$L_{rot}^{(k)} = \arccos(\overline{z}_k\cdot\widehat{z}_k), \tag{8}$$

where $z^* = [2(x^*z^* + w^*y^*), 2(y^*z^* - w^*x^*), z^{*2} + w^{*2} - y^{*2} - x^{*2}]$. So an instance can be supervised with:

$$L_{ins}^{(k)} = \alpha L_{cen}^{(k)} + \beta L_{rot}^{(k)} + \gamma L_{sem}^{(k)}, \tag{9}$$

where α, β, and γ are the hyper-parameters to balance the magnitude. A proposal can be evaluated by the confidence score c_k with the loss function:

$$L_{conf}^{(k)} = L_{ins}^{(k)}\cdot c_k - \log(c_k). \tag{10}$$

Thus, the proposed loss function is given as:

$$L = \frac{1}{M}\sum_{i=1}^{M} L_{vote}^{(i)} + \frac{1}{M_{\mathbb{I}}}\sum_{k=1}^{K}\left(L_{ins}^{(k)} + L_{conf}^{(k)}\right)\mathbb{I}(k), \tag{11}$$

where M represents the number of the predicted centers, K is the instance number, $M_{\mathbb{I}}$ is the number of the positive proposal mask, and $\mathbb{I}(k)$ denotes an indicator function of the positive proposal mask.

3.5. Implementation details

Input data. The input to the proposed network is a point cloud of N = 20k points randomly sampled from a scene which is collected by the sensor of PhoXi 3D Scanner M and is executed after the background subtraction. In addition to the xyz coordinate information, each point contains a piece of normal information that is computed by the principal component analysis algorithm with a non-iterative method to extract the eigenvector from the covariance matrix implemented by Open3D [51].

Network architecture detail. The backbone framework is based on PointNet++ [29] with four SA layers and two feature propagation (FP) layers. The SA layers have the increased receptive field radius of r, $2r$, $3r$, and $6r$ in four layers where r is usually set to 10 mm or 15 mm based on the object scales. The sampling number of the SA layers are 2048, 1024, 512, and 256 points, respectively. The FP layers up-sample from 256 points back to $M = 1024$ votes with $F(256)$-dimensional features and 3D coordinates. The voting module is implemented by an MLP of three layers with output dimensions of 256, 256, and 259 including extra 3D coordinates.

The votes are voxelized in the 3D Euclidean space with a general step of 5 mm to build a heatmap in the proposal module. Then the map is smoothed by a Gaussian filter with a kernel size of $3 \times 3 \times 3$. The top K = 64 local peaks are considered proposals and only the peaks over the threshold of 0.4 are accepted as valid proposals in the inference stage. For each proposal, the $H = 32$ votes are gathered in a voting field of general radius r.

The attention module is implemented by two branches of shared MLP with the same structure. Each branch has three layers with the output dimensions of 128, 128, and 128 separately for the features and the attention weights. The fused instance features also have 128 dimensions. Finally, the instance features are fed into a head MLP with the output dimensions of 128, 128, and $3 + 4+NC+1$ which consists of 3 center regression values, 4 rotation regression values, NC number of classes for semantic classification, and a confidence score. The output can be directly adopted as the inference results for evaluation or application without any post-processing steps like NMS or ICP.

Training of the network. Our network is prototyped with PyTorch 1.5.0 on an NVIDIA RTX 3080 GPU and is trained with the Adam optimizer, batch size of 8, and initial learning rate of 0.001. The learning rate decays by a factor of 0.1 after 5 epochs and then by another 0.1 after 10 epochs for the Fraunhofer IPA dataset [52]. The forward-pass time of the network with the input size of 20,000 $\times$ 6 is about 65 milliseconds for a single frame.

4. Experiments

We evaluate the proposed network with the public Fraunhofer IPA bin-picking dataset [52] as the benchmark, the synthetic and real-world datasets including three categories of industrial parts are generated for the bin-picking experiments.

4.1. Datasets and evaluation metrics

Benchmark dataset. As presented in Table 1, the public Fraunhofer IPA bin-picking dataset [52] is utilized to evaluate the performance of the network. This dataset contains eight objects from the Siléane dataset [53]: Bunny, Candlestick (C.Stick), Pepper, Brick, Gear, T-Less 20, T-Less 22, T-Less 29 and two objects from the real-world data: Ring screw and Gear shaft, as displayed in Fig. 5. The training and testing datasets of the eight Siléane objects are the simulated data. The training sets of two real-world objects are the simulated data and the testing sets are obtained from the real-world data. The synthetic testing sets of two real-world objects are also available, but the depth image and the label of the Ring screw seem to be incompatible. The information about all objects is summarized in Table 1. Each scene only takes one kind of object. In a cycle, the object is dropped randomly into a bin one by one from zero to its drop limit and each drop is captured as a scene. Each cycle contains scenes from an empty background scene to objects of drop limit. We take the drop number from 5 to the drop limit to keep the complexity of the scene.

Bin-picking dataset of industrial parts. We generate a synthetic dataset of three kinds of industrial parts for the real-world bin-picking experiments. The dataset is challenging for pose estimation because of the local features and the rotation ambiguities of the stacked industrial parts. The physical simulation is constructed under the environment of NVIDIA PhysX and is rendered under the environment of OpenGL. The data are collected for training and testing with 28,000 and 2000 scenes, respectively. Each scene contains more than ten objects including all categories. Table 2 provides the information and the model illustration of three kinds of industrial parts. The synthetic scenes and the corresponding instance labels are given in Fig. 6.

Evaluation metric. The Fraunhofer IPA dataset adopts the evaluation metric proposed by Brégier *et al.* [53,54] which defines a representation of pose $\mathscr{P}$ as a finite set of points $\mathscr{R}(\mathscr{P})$ with at most 12 dimensions, where the size and dimension of the settings depend on the categories of the object symmetry. This representation is suitable for the rigid object poses since its information on symmetry can be obtained from the principal component analysis of model point clouds. So, the metric of two poses defined by the minimum Euclidean distance between point sets can be evaluated as:

Table 1
Fraunhofer IPA bin-picking dataset [52].

Object	Diameter (mm)	Symmetry	Training drop limit	Training cycles	Training scenes	Testing drop limit	Testing cycles	Testing scenes
Bunny	234	No	80	500	38,000	80	10	760
C.Stick	191	Revolution	60	500	28,000	60	10	560
Pepper	335	Revolution	90	500	43,000	90	10	860
Brick	37	Cyclicity	150	500	73,000	150	10	1460
Gear	121	Revolution	60	500	28,000	60	10	560
T-Less 20	107	Cyclicity	99	500	47,500	99	10	950
T-Less 22	108	No	100	500	48,000	100	10	960
T-Less 29	135	Cyclicity	79	500	37,500	79	10	750
Ring screw	155	Cyclicity	35	500	15,500	28	10	240 (Real)
Gear shaft	437	Revolution	30	500	13,000	22	10	180 (Real)

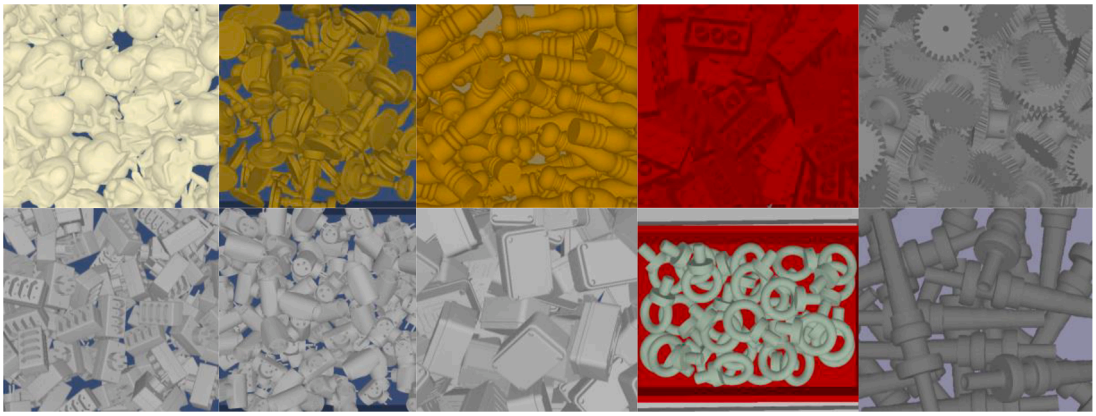


Fig. 5. Stacked scenes of the Fraunhofer IPA dataset [52].

Table 2
Bin-picking dataset of three kinds of industrial parts.

Object	Diameter (mm)	Symmetry	Geometric model
Vertical bearing pedestal	165	No	
Connecting rod	143	Cyclicity	
Slider bearing pedestal	125	Cyclicity	

$$d(\mathscr{P}_1, \mathscr{P}_2) = \min_{p_1 \in \mathscr{R}(\mathscr{P}_1), p_2 \in \mathscr{R}(\mathscr{P}_2)} \| p_2 - p_1 \| . \tag{12}$$

A pose hypothesis is considered as precision if its distance from the ground-truth pose is less than a threshold of 10 % × D, where D is the diameter of the object's minimum bounding sphere. Then, the performance is measured by average precision (AP), which denotes the area under the precision-recall (PR) curve [55]. In our method, the metric takes the predicted confidence score c_k as the sort criteria to make compromises between precision and recall. Following the unified standard [52,53], only the objects with occlusion of less than 50 % are taken as the retrieving goals on the Fraunhofer IPA dataset.

For the bin-picking dataset of industrial parts, another evaluation metric based on the average distance of model points (ADD) [56] is adopted because the PR curve per object may neglect the uniformity of one scene with more than one object and cannot reveal the contrast among objects. This ADD metric defines the pose error of the model $\mathscr{M}$ as:

$$d(\mathscr{P}_1, \mathscr{P}_2) = \underset{x \in \mathscr{M}}{avg} \| \mathscr{P}_1(x) - \mathscr{P}_2(x) \| . \tag{13}$$

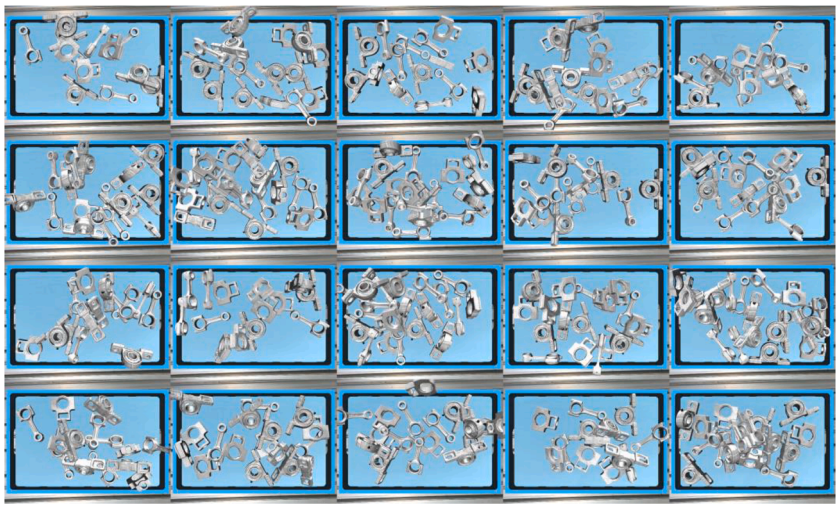
In the pose estimation task, it is usually considered as true positive (TP) if the error is less than a threshold of 10 % × D and the precision takes the ADD as the evaluation metric [56]. Besides, for multi-object detection, recall is also an important criterion. Thus, the precision and recall values are used to evaluate the network on the bin-picking dataset of industrial parts.

4.2. Experimental results

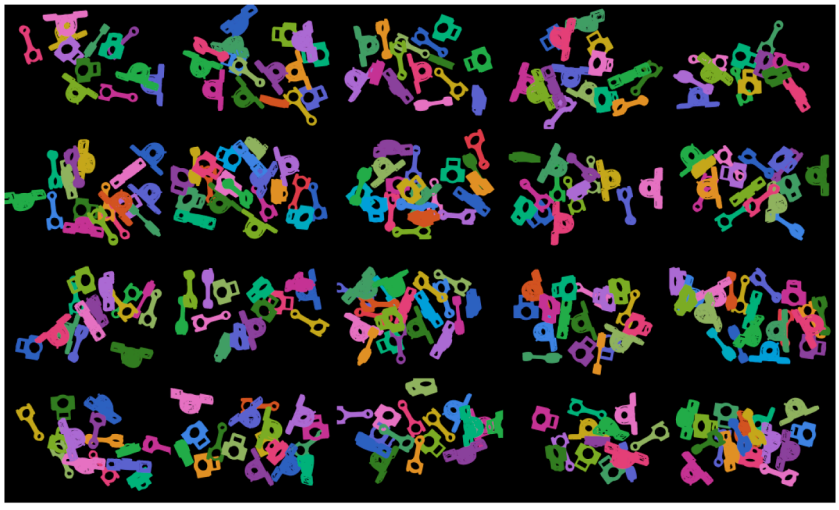
Results on the Fraunhofer IPA dataset. Pose estimation results on the Fraunhofer IPA dataset are presented in Fig. 7, and Table 3 lists the AP performance results of our method and some other pose estimation methods based on point clouds on the Fraunhofer IPA dataset. There are some traditional methods without learning such as PPF [35] and LINEMOD [56]. These methods cannot afford to deal with complex scenes and present unideal results. Others are the learning-based methods including the state-of-the-art methods OP-Net [57] and PPR-Net++ [46] which are considered the main baseline.

The results of these methods listed in [46] and [53] are employed. It should be noted that the Brick AP value of 0.47 of PPR-Net++ is likely to be a literal typo since its AP value reaches a significantly better level of 0.93 in its analysis experiment. Thus, we replace it with 0.93 in Table 3 (marked with *). In our method, some parameters are set up respectively for each object to adapt to scale effects. These parameters are mainly scale-related, such as receptive field radius and voxel size, and thus can be intuitively adjusted based on the object model diameter.

On the synthetic Siléane dataset of eight objects, the proposed method performs better than PPR-Net++ in general. Specifically, our method reaches an improvement of 2 % to 6 % on six objects and the other two objects with nearly full AP values are basically the same. The



(a) RGB images of the synthetic dataset.



(b) Instance labels of the synthetic dataset.

Fig. 6. The synthetic dataset of three kinds of industrial parts. (a) RGB images of the synthetic dataset. (b) Instance labels of the synthetic dataset.

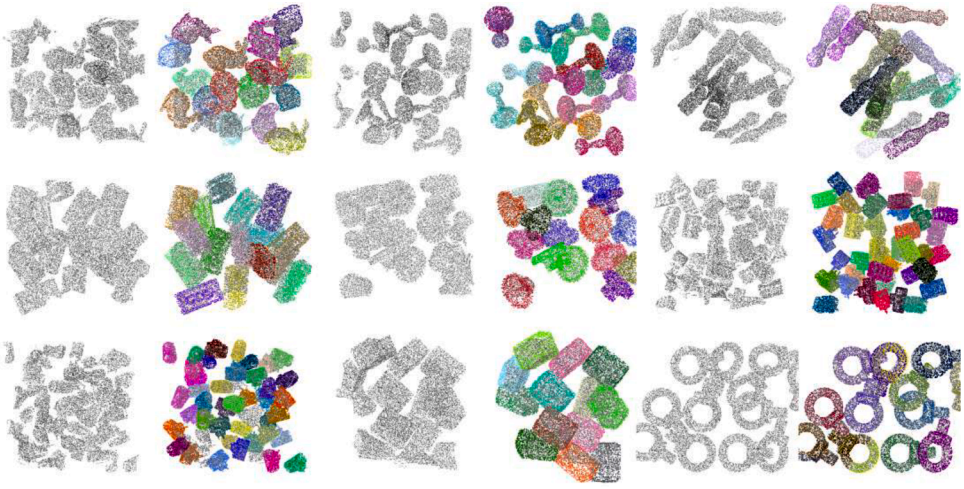


Fig. 7. Pose estimation results on the Fraunhofer IPA dataset.

Table 3
Performances of pose estimation on the Fraunhofer IPA dataset with metric of 10 % × *D*.

Object	Bunny	C.Stick	Pepper	Brick	Gear	T-Less 20	T-Less 22	T-Less 29	Ring screw	Gear shaft
PPF [35]	0.29	0.16	0.06	0.08	0.62	0.20	0.08	0.19	–	–
PPF PP [35]	0.37	0.22	0.12	0.13	0.63	0.23	0.12	0.23	–	–
LINEMOD+ [56]	0.39	0.38	0.04	0.31	0.44	0.25	0.19	0.20	–	–
LINEMOD+ PP [56]	0.45	0.49	0.03	0.39	0.50	0.31	0.21	0.26	–	–
Sock *et al.* [58]	0.74	0.64	0.43	–	–	–	–	–	–	–
OP-Net L_{ori1} PP [57]	0.94	0.97	0.98	0.42	0.84	0.88	0.86	0.58	0.93	0.99
OP-Net L_{ori2} PP [57]	0.76	0.96	0.93	0.80	0.60	0.58	0.55	0.39	0.75	**1.00**
PPR-Net++ [46]	**0.99**	0.98	0.98	0.93*	**1.00**	0.93	0.92	0.94	**0.98**	0.99
Ours	**0.99**	**1.00**	**1.00**	**0.97**	**1.00**	**0.99**	**0.94**	**0.98**	0.85	0.98

AP values of the proposed method can reach 1.00 on three revolution objects of C.Stick, Pepper, and Gear. Besides, the performance results of objects with cyclic symmetry including Brick, T-Less 20, and T-Less 29 are improved by a relatively high increase.

On the real-world Fraunhofer IPA dataset, the proposed method exhibits considerable difference between the two objects. For the Gear shaft, the AP value of our method is lower than the baseline in a limited gap. However, our method fails to deal with the Ring screw with a general setting. In the training phase, our method keeps an ideal performance and the AP value can reach 0.98 on the training dataset. However, the low level of the AP value is 0.093 in the testing phase. This can be considered an overfitting phenomenon against the noisy real-world testing dataset. Furthermore, the synthetic training dataset of Ring screw has no noise and the number of the training datasets is small.

Therefore, we further add Gaussian noise to each point in the training phase to simulate the sensor noise. As presented in Table 4, the AP value of the Ring screw is improved when the noise scale is set to 2 mm. Table 4 also shows the noise influence on other objects including the synthetic testing dataset (T-Less 20) and the real-world testing dataset (Gear shaft).

Fig. 8 plots the trend of the AP value with distance threshold from 0 to 10 % × *D* on Brick, Bunny, C.Stick, and Pepper of the Fraunhofer IPA dataset. According to the curve, our method can maintain a high level when the distance threshold is larger than 5 % × *D*. Table 5 further shows an accurate comparison with a distance threshold of 2 % × *D*. With a stricter metric of 2 % × *D*, the AP values of PPR-Net++ and our method are different on four objects. For Brick, PPR-Net++ shows slightly worse performance and our method improves by about 30 %. For Bunny, PPR-Net++ shows better performance and our method drops by about 55 %. For C.Stick and Pepper, both methods show better performance. Our method drops by 8 % on C.Stick and improves by 2 % on Pepper.

Results on the synthetic and real-world datasets of industrial parts. Fig. 9 gives the pose estimation results on the synthetic dataset of industrial parts. It can be observed that the overall pose results are very accurate, and a small number of results have certain deviations. The pose results around the hole of the workpiece produces a rotation deviation (in the red circle region of Fig. 9), which can be considered as being affected by the rotation similarity of the round hole feature. Fig. 10 displays the pose estimation results of the real-world dataset of industrial parts, in which the manually annotated labels are denoted by yellow point clouds. It can be observed that the performance results of the real-world dataset are slightly worse than the synthetic results. For both bearing pedestals, there will be small rotation deviation, but the overall performance accuracy still maintains at a high level and meets the grasping requirements. It also still gives relatively accurate predictions for objects that are difficult to be annotated.

Table 6 shows the performance results of the proposed method on the synthetic and real-world datasets of industrial parts. The labels of the synthetic dataset are exported by the physical engine, and a total of 2000 testing scenes are generated. The real-world dataset is manually annotated, and there are 100 testing scenes in total. The underlying parts with a large portion of occlusion, that cannot be accurately annotated, are not marked. Thus, this class of unlabeled objects is not included in the statistics for performance estimation.

Table 4
Effects of noise scales during training phase on the Fraunhofer IPA dataset.

Noise scale (mm)	0	1	2	3
T-Less 20	**0.986**	0.983	0.979	0.124
Gear shaft	**0.980**	0.832	0.925	0.969
Ring screw	0.093	0.409	**0.849**	0.415

4.3. Experimental discussions

In this section, the performance results of the proposed modules are evaluated by adjusting some concerned hyper-parameters or replacing some alternative methods as a comparison. Some results are evaluated on the Fraunhofer IPA dataset and others on the bin-picking dataset of industrial parts on different analysis modules.

Peak threshold. The threshold related to the heat peak voxels mentioned in Section 3.2 directly determines the quality of accepted proposals. If the threshold is too small, many inferior or invalid proposals may be propagated forward and thus reduce the precision. On the contrary, few proposals are adopted and may pull down the recall value if the threshold is too big. In order to test the effect of the threshold on different object sizes and voxel sizes, the objects of different sizes (from 37 mm to 335 mm) are selected from the Fraunhofer IPA dataset which should be set to different voxel sizes (from 3 mm to 7 mm). Fig. 11 shows the trend of the AP values with different peak thresholds for Brick, Bunny, C.Stick, and Pepper from the Fraunhofer IPA dataset. In the threshold range from 0.3 to 0.6, the curve keeps a stable trend with the desired performance. Thus, the threshold range mentioned above amounts to about 5–10 votes as a center voxel weight of 0.0923 and the neighbor voxel weights of 0.0560 for the normalized 3D Gaussian filter are adopted. This means that the peak threshold does not have a significant impact on the AP values when the crucial hyper-parameter is selected within a reasonable range.

Rotation loss function. This section discusses the quaternion loss of orientation increment (Eq. (6), spherical linear interpolation denoted by Slerp version) and the minimum rotation angle (Eq. (7) denoted by Minra version) mentioned in Section 3.4. The normalized linear interpolation (Nlerp version) is taken as a baseline. As shown in Table 7, the performance results of the Minra version are a little worse than those of the Slerp version. This can be explained by the backpropagation process. For the Minra loss, the training target of $|\overline{q}_k \cdot \widehat{q}_k|$ is 1. Notice that both $\overline{q}_k$ and $\widehat{q}_k$ are unit quaternions, so $\overline{q}_k$ is fitted to $\widehat{q}_k$ when $\overline{q}_k \cdot \widehat{q}_k > 0$ and is fitted to $-\widehat{q}_k$ when $\overline{q}_k \cdot \widehat{q}_k < 0$ essentially. However, $\widehat{q}_k$ and $-\widehat{q}_k$ are equivalent for the rotation representation, which may lead to ambiguity for the network learning.

Feature fusion module. Table 8 compares the performance results of several feature fusion modules mentioned in Section 3.3 to evaluate our proposed attention module on the synthetic bin-picking dataset of

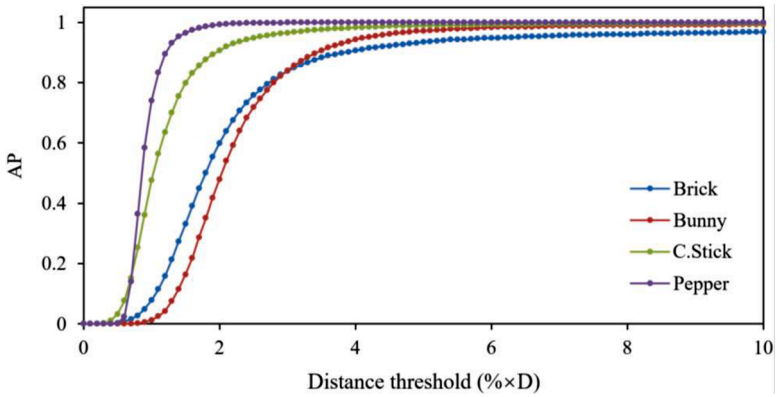


Fig. 8. Curves of AP value and distance threshold on the Fraunhofer IPA dataset.

Table 5
Performance comparison with stricter metric on the Fraunhofer IPA dataset.

Distance threshold	Method	Brick	Bunny	C.Stick	Pepper
10 % × D	PPR-Net	0.908	**0.997**	0.966	0.996
	PPR-Net++	0.930	**0.997**	**0.996**	0.995
	Ours	**0.967**	0.992	**0.996**	**1.000**
2 % × D	PPR-Net	0.249	0.931	0.953	0.962
	PPR-Net++	0.290	**0.943**	**0.986**	0.974
	Ours	**0.599**	0.480	0.906	**0.992**

industrial parts. PN denotes the pipeline using the shared MLP module of the PointNet for feature fusion. AMX denotes the use of the attention module of the matrix version and AML denotes the use of the attention module of the shared MLP version. Table 8 shows that the values of the precision and recall are roughly equal with a metric of 10 % × *D,* and AML shows improvement by about 3 % on both the values of the precision and recall with a stricter metric of 2 % × *D*. AMX shows a better performance only with the metric of 5 % × *D*. Actually, AML shows more improvement in the pipeline without the heatmap proposal strategy, because PN needs a refinement step of ICP for better performance. Considering the practical applications, higher performance with a strict metric can abandon the post-refinement and achieve a higher robotic manipulation accuracy and efficiency. In this way, AML is finally chosen for the proposed pipeline of the pose estimation.

Voting radius. For the method based on the Hough voting, the receptive field determined by the voting radius in the proposed network is a crucial factor to transform the discrete data into the quantization indicator. Table 9 summarizes the effects of the voting radius in the training and testing phases on the bin-picking dataset of industrial parts to evaluate the compromise for multiple objects in a single scene. From Table 9, it can be observed that the voting radius mainly affects the precision because it decides the quality of the adopted votes. On the other hand, it has almost no effect on recall. This benefits from the voting center generated by the proposal module. Besides, we find that a bigger voting radius is beneficial to the warm-up of the network model in the early training phase and a smaller voting radius can be chosen to extract more valuable votes for feature fusion in the testing phase.

4.4. Robotic bin-picking experiments

As illustrated in Fig. 12, the experimental platform is constructed by Universal Robots UR5 and PhoXi 3D Scanner M. Three kinds of industrial parts, the surface features of which are homogeneous textureless, are adopted to predict the poses for the robotic bin-picking experiments in the cluttered and occluded scenes. The hardware communications and the data information interactions between sensor, robot and host computer are presented in Fig. 13. In the offline stage, the hand-eye calibration is implemented to obtain the transformation relationship between the robot and sensor coordinate systems, the reader is referred to [59] for an in-depth presentation of the hand-eye calibration. To avoid replacing fixtures, the electric gripper with a parallel two-finger structure is chosen for picking up the industrial parts, and the

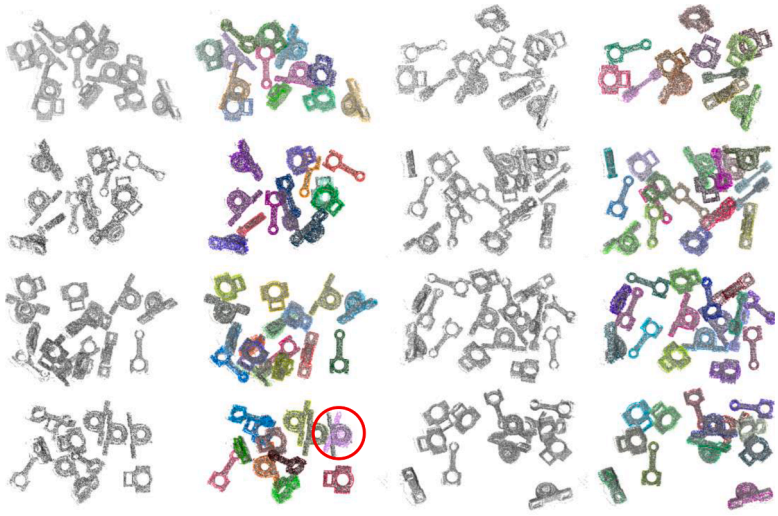


Fig. 9. Pose estimation results on the synthetic dataset of industrial parts.

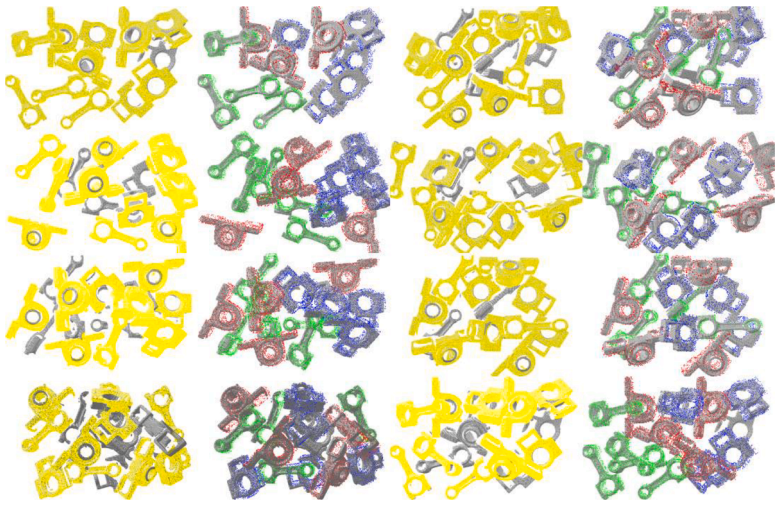


Fig. 10. Pose estimation results on the real-world dataset of industrial parts.

Table 6
Performances of pose estimation on the datasets of industrial parts.

Object	Synthetic dataset		Real-world dataset	
	Precision	Recall	Precision	Recall
Vertical bearing pedestal	0.970	0.986	0.943	0.951
Connecting rod	0.964	0.976	0.997	0.997
Slider bearing pedestal	0.970	0.982	0.959	0.957
Mean	0.968	0.982	0.966	0.968

manipulating positions and postures are specially assigned to different industrial parts, as shown in Fig. 14.

The pose estimation process and the grasping manipulation are described in Fig. 15, each column of which displays input point cloud, heatmap, pose estimation results, selected part, and actual scene of grasping manipulation, respectively. According to the experimental results, the proposed network model of this article can accurately estimate the poses of all upper industrial parts in real physical scenes. The two-finger gripper approaches the selected part along the z-axis direction of the grasping pose to avoid collisions with other parts as much as possible. Fig. 16 presents the whole grasping process in a real-world physical scene with clutter and occlusion. It should be noted that Photoneoprovides a software of Bin Picking Studio installed on Vision Controller for robotic intelligence operation [60]. To perform the comparison and confirm the performance, some crucial test results of Vision Controller are extracted from the relevant websites. According to the test results from the official website of Photoneo, the success rate of bin-picking is 99.9 %, and the cycle time of performing pick-and-place operation for one object is 5 s, which can also be counted from the videos provided on the Photoneo's website. Furthermore, for practical industrial applications, Photoneoprovides the rapid deployment software for at least seven brands of robots, such as ABB, Fanuc, Kawasaki and Universal Robots. The setup time of industrial deployment can be controlled within 20 min to realize the robotic bin-picking manipulation. In this work, a laboratory-developed robot bin-picking platform is constructed to verify the pose prediction performance. The total gripping time of per industrial part is about 28 s. The entire time-consuming process mainly consists of point cloud acquisition, pose estimation, and robot movement. It must be admitted that compared with Photoneo, the fixture design and electrical design of the robotic bin-picking experimental platform have not yet reached the industry-level standards. Therefore, to ensure the stability at the moment of picking operation, the approaching speed of the gripper is reduced as the gripper approaches the selected object. At the same time, the movement speed of the robot is limited to avoid the falling of the picked part during the pick-and-place process. The 100 % success rate is achieved when the error of the pose

Table 7
Performances of the quaternion linear interpolation on the Fraunhofer IPA dataset.

Loss function	Bunny (No symmetry)		T-Less 20 (Cyclicity)	
	10 % × D	*2 % × D*	*10 % × D*	*2 % × D*
Slerp version	**0.992**	**0.480**	**0.986**	0.107
Minra version	0.991	0.001	0.982	**0.243**
Nlerp version	0.991	0.076	0.030	0.000

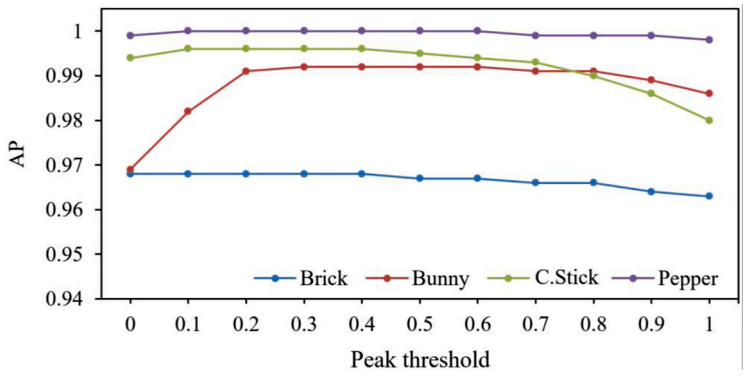


Fig. 11. Peak threshold-AP curves on the Fraunhofer IPA dataset.

Table 8

Distance threshold	*2 % × D*			*5 % × D*			*10 % × D*		
Module	PN	AMX	AML	PN	AMX	AML	PN	AMX	AML
(a). Performances of the feature fusion module on the dataset of industrial parts (Precision).									
Vertical bearing pedestal	0.412	0.398	**0.441**	**0.941**	**0.941**	0.938	**0.972**	0.969	0.970
Connecting rod	0.498	0.531	**0.548**	0.926	**0.937**	0.936	0.959	0.963	**0.964**
Slider bearing pedestal	0.238	**0.289**	0.266	0.929	**0.933**	0.928	0.969	0.969	**0.970**
Mean	0.383	0.379	**0.419**	0.932	**0.937**	0.934	0.967	0.967	**0.968**
(b) Performances of the feature fusion module on the dataset of industrial parts (Recall).									
Vertical bearing pedestal	0.636	0.625	**0.659**	0.965	**0.968**	0.965	0.984	0.985	**0.986**
Connecting rod	0.691	0.717	**0.730**	0.949	**0.957**	**0.957**	0.970	0.975	**0.976**
Slider bearing pedestal	0.479	0.449	**0.505**	0.955	**0.959**	0.955	0.982	**0.983**	0.982
Mean	0.602	0.597	**0.631**	0.956	**0.961**	0.959	0.979	0.981	**0.982**

Table 9

Training radius (mm)	30			20		10		5
Testing radius (mm)	30	20	10	20	10	10	5	5
(a). Effects of voting radius on the dataset of industrial parts (Precision)								
Vertical bearing pedestal	0.952	0.966	0.950	0.949	**0.971**	0.970	0.965	0.956
Connecting rod	0.939	0.955	0.943	0.938	0.956	**0.964**	0.961	0.940
Slider bearing pedestal	0.954	0.962	0.954	0.949	0.964	**0.970**	0.964	0.950
Mean	0.948	0.961	0.949	0.945	0.964	**0.968**	0.963	0.949
(b). Effects of voting radius on the dataset of industrial parts (Recall).								
Training radius (mm)	30			20		10		5
Testing radius (mm)	30	20	10	20	10	10	5	5
Vertical bearing pedestal	0.986	0.984	0.969	**0.987**	0.982	0.986	0.980	0.981
Connecting rod	**0.978**	0.975	0.955	0.975	0.969	0.976	0.970	0.971
Slider bearing pedestal	0.982	0.981	0.966	**0.983**	0.979	**0.983**	0.974	0.979
Mean	**0.982**	0.980	0.963	**0.982**	0.977	**0.982**	0.975	0.977

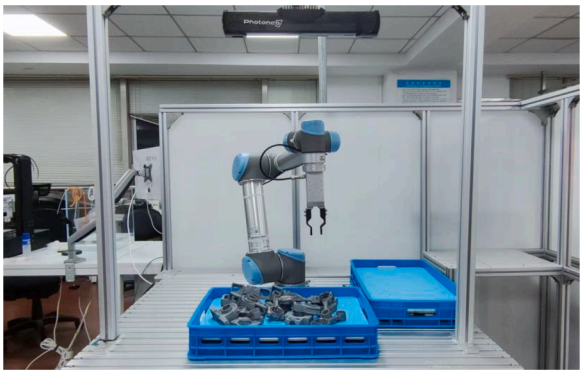


Fig. 12. The experimental scene and the hardware configuration for the robotic bin-picking manipulation of industrial parts in the cluttered and occluded scene.

prediction is less than 10 % of the diameter of the 3D model and the collision detection is considered in the process of robotic manipulation. The statistical results of the success rate do not consider these special cases, such as the clamping failure of the electric gripper and the communication failure between the host computer and the PhoXi 3D Scanner. The Universal Robots UR5 is used to perform the pick-and-place operation. The deployment between PhoXi 3D Scanner, UR5, and host computer is only for performance evaluation, and has not been optimized for practical industrial applications. Thus, the deployment or setup time is much higher than 20 min. According to the analysis above, the whole manipulation time of this work can be reduced by improving the quality of the fixture, optimizing the path of the robot, and increasing the movement speed of the robot which are also the focuses of our future work. The video of a robotic bin-picking experiment is also provided on the relevant webpage associated with this article.

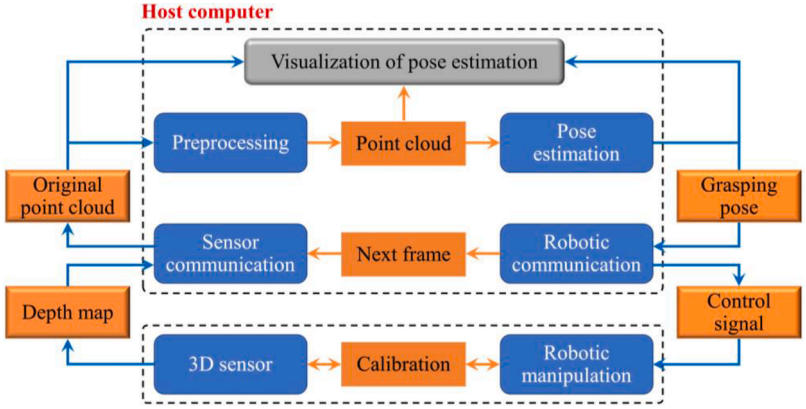


Fig. 13. Workflow of sensor, robot, and host computer for the robotic bin-picking experiments.

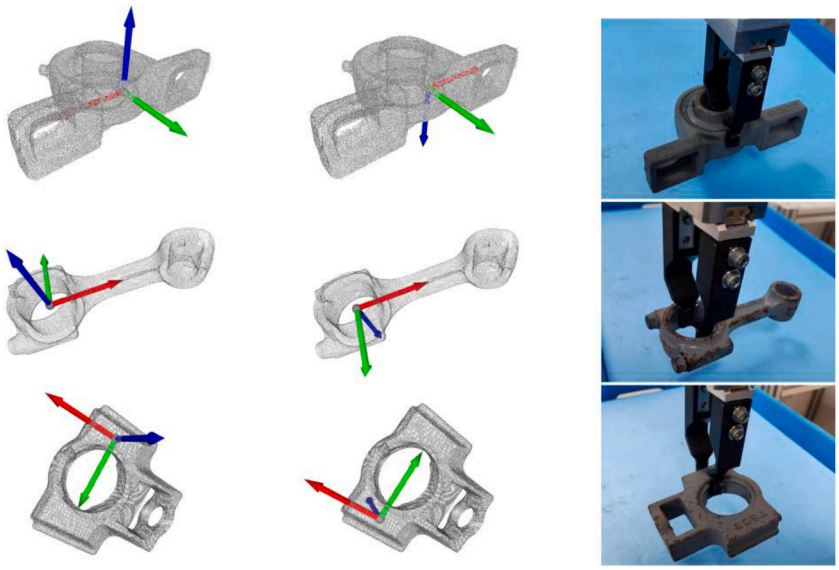


Fig. 14. Pre-settings of the grasping poses for three kinds of industrial parts including vertical bearing pedestal, connecting rod, and slider bearing pedestal.

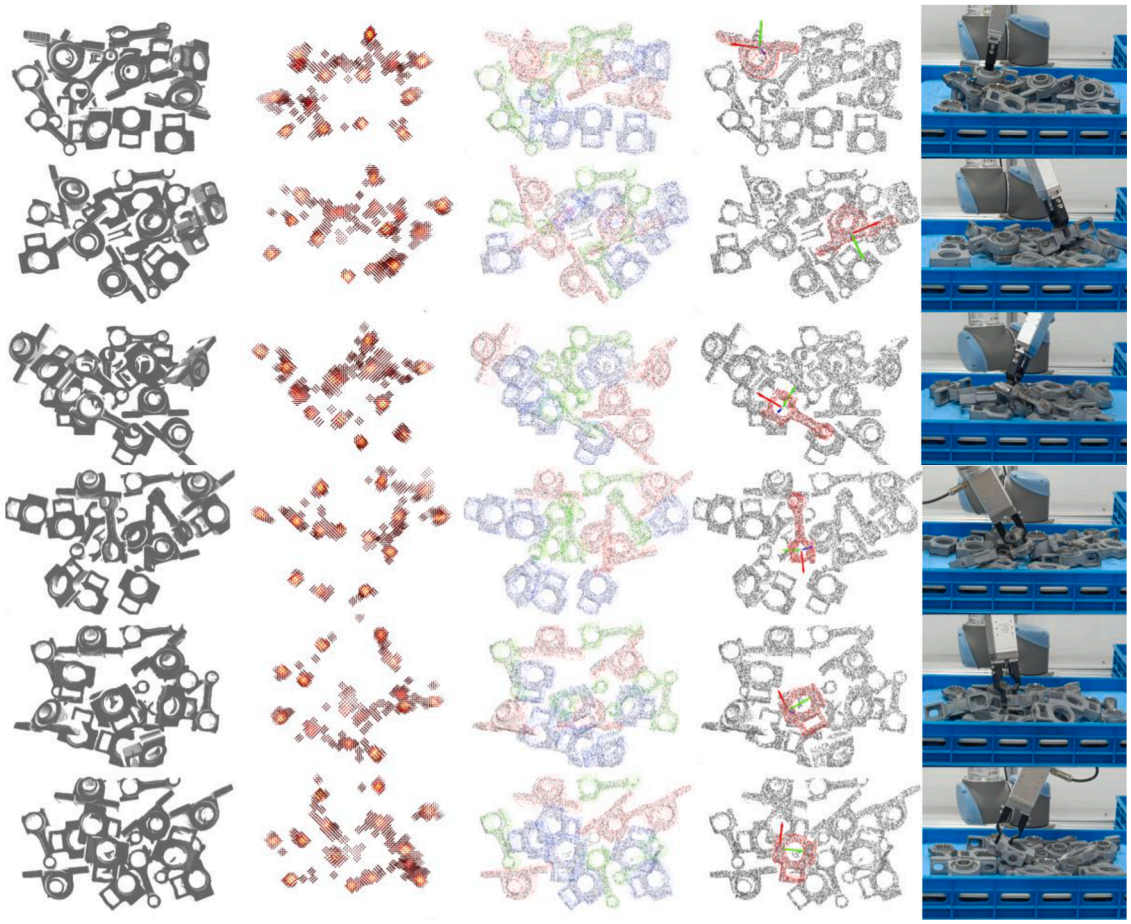


Fig. 15. Robotic bin-picking demonstration of industrial parts in the real-world scenes with clutter and occlusion. The input point cloud, heatmap, pose estimation results, selected part, and grasping manipulation are presented in each column, respectively.

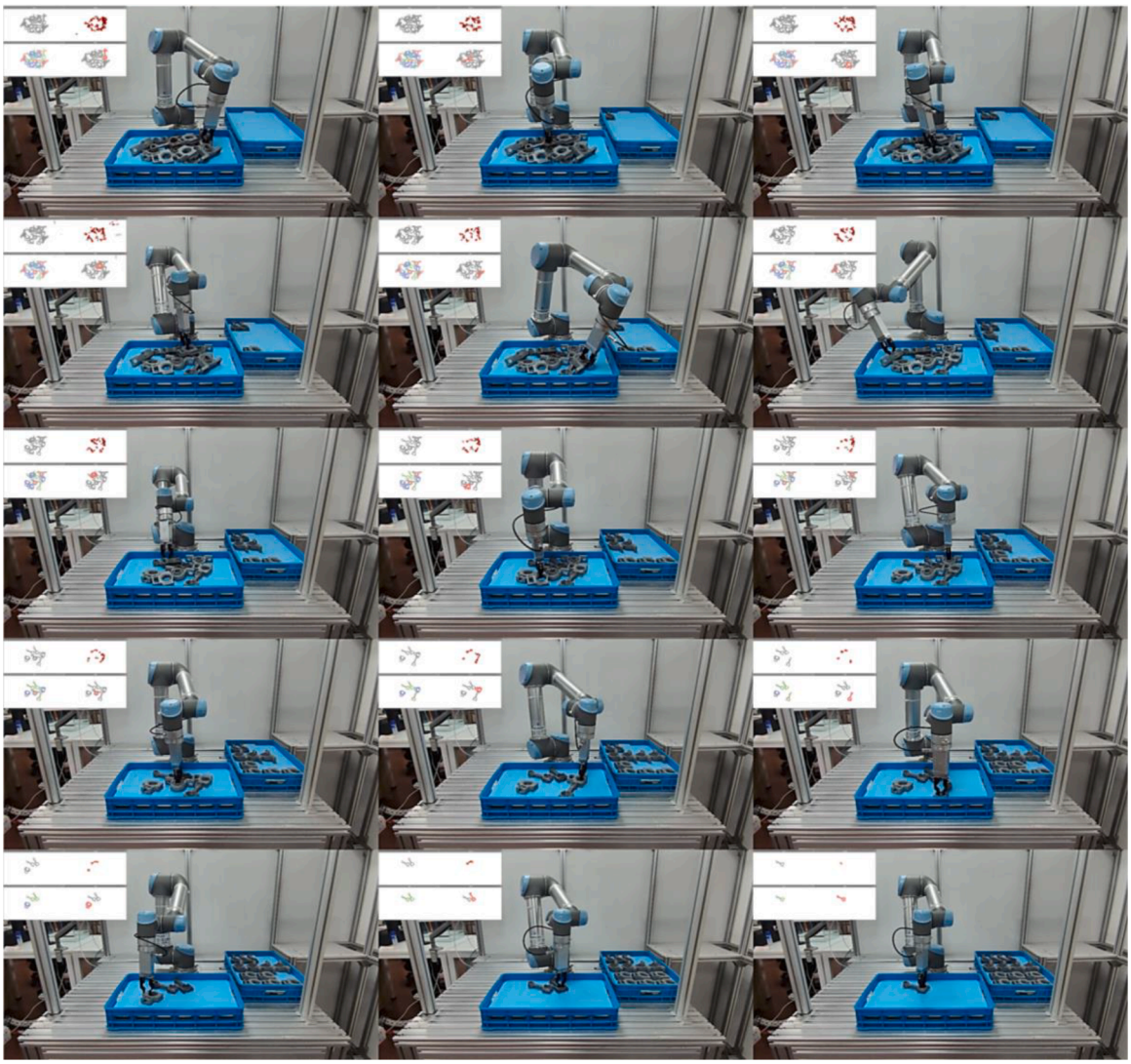


Fig. 16. The whole process of the robotic bin-picking experiment for three kinds of industrial parts in a cluttered and occluded scene.

5. Conclusion

This article deals with the subject of the 6D pose estimation on point cloud. An end-to-end network is proposed to predict the poses of textureless objects for the industrial bin-picking scenarios. The network model learns from point-wise features to instance-wise features and thus generates pose predictions corresponding to different objects. The unsegmented scene point cloud is directly taken as the input of the network. A rough voting module based on the 3D heatmap is designed for the center prediction of each instance. After learning the voting centers and voxelizing the votes, the Gaussian filter is adopted to smooth the voxelized votes for generating the heatmap of voting centers. Then, the center proposals of object instances are analyzed and suggested in terms of the local peaks of the heatmap. The relevant features of votes for each proposed instance can be obtained by utilizing the proposals from the heatmap. A feature fusion module based on the attention mechanism is constructed to process the gathered instance features. By learning the weight of each voting feature, the point-wise features can be adaptively fused into the instance-wise features for pose estimation. In the stage of supervising poses, the quaternion is used to represent the rotation and the spherical linear interpolation of the quaternion is adopted to calculate the rotation angle increment as the loss for regression. In order to address the problem of multiple ground-truth poses caused by cyclic symmetry, the rotation angle increments from the predicted pose to each labeled pose are compared and the closest label is selected for regression. The better performance of the proposed network is achieved on the public Fraunhofer IPA dataset as well as the synthetic and real-world datasets of industrial parts. Finally, using the predicted pose results, the experiments of the robotic bin-picking manipulation are implemented on the real-world scenes of industrial parts, which further illustrates the effectiveness of the end-to-end pose estimation network.

It should be noted that deep learning methods usually rely on a large amount of data for training the network in order to achieve accurate pose predictions in unknown scenes. Similarly, the proposed deep learning-based pose estimation framework suffers from some limitations as the traditional deep learning methods. Although the proposed pipeline of this article does not require the geometric models of objects during the training process, it still requires a substantial amount of scene data as well as the corresponding label information of the existing models in the scene point cloud. In addition, the proposed network cannot directly handle the unknown objects that may appear in the scene, and usually requires the generation of a new dataset specifically for training on those objects. The PPFs-based pose estimation method that depends on the feature matching is characterized by a low-cost training procedure compared to network models. Thus, it is valuable to combine the local feature description with the deep learning method to alleviate the training cost for the new types of objects that they have

never been seen before, which deserves further investigation. Apart from mentioned pose estimation, the issues related to the collision detection and path planning, which are crucial for completing a comprehensive robotic bin-picking task, in the process of the pick-and-place operation would be the topics of particular interest in future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source codes with the synthetic and real-world datasets of industrial parts that support the findings of this study are available on email request from the corresponding author.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 52275500). The authors cordially appreciate the reviewers for their insightful comments and hard work.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.rcim.2023.102671.

References

[1] K.N. Kaipa, A.S. Kankanhalli-Nagendra, N.B. Kumbla, S. Shriyam, S.S. Thevendria-Karthic, J.A. Marvel, S.K. Gupta, Addressing perception uncertainty induced failure modes in robotic bin-picking, Robot. Comput.-Integr. Manuf. 42 (2016) 17–38, https://doi.org/10.1016/j.rcim.2016.05.002.

[2] S. D'Avella, C.A. Avizzano, P. Tripicchio, ROS-industrial based robotic cell for Industry 4.0: eye-in-hand stereo camera and visual servoing for flexible, fast, and accurate picking and hooking in the production line, Robot. Comput.-Integr. Manuf. 80 (2023), 102453, https://doi.org/10.1016/j.rcim.2022.102453.

[3] B. Tipary, G. Erdős, Generic development methodology for flexible robotic pick-and-place workcells based on digital twin, Robot. Comput.-Integr. Manuf. 71 (2021), 102140, https://doi.org/10.1016/j.rcim.2021.102140.

[4] Y.Z. Jiang, Z.Z. Huang, B. Yang, W.Y. Yang, A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation, Robot. Comput.-Integr. Manuf. 78 (2022), 102366, https://doi.org/10.1016/j.rcim.2022.102366.

[5] M.Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T.K. Marks, R. Chellappa, Fast object localization and pose estimation in heavy clutter for robotic bin picking, Int. J. Rob. Res. 31 (2012) 951–973, https://doi.org/10.1177/0278364911436018.

[6] D. Buchholz, Bin-Picking: New Approach for a Classical Problem, Springer, Braunschweig, Germany, 2016, https://doi.org/10.1007/978-3-319-26500-1.

[7] C.G. Zhuang, S.F. Li, H. Ding, Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking, Robot. Comput.-Integr. Manuf. 82 (2023), 102541, https://doi.org/10.1016/j.rcim.2023.102541.

[8] C. Wang, D.F. Xu, Y.K. Zhu, R. Martín-Martín, C.W. Lu, F.F. Li, S. Savarese, DenseFusion: 6D object pose estimation by iterative dense fusion, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, California, USA, 2019, pp. 3338–3347, https://doi.org/10.1109/CVPR.2019.00346.

[9] Y.S. He, H.B. Huang, H.Q. Fan, Q.F. Chen, J. Sun, FFB6D: a full flow bidirectional fusion network for 6D pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Tennessee, USA, 2021, pp. 3002–3012, https://doi.org/10.1109/CVPR46437.2021.00302.

[10] C. Sahin, G. Garcia-Hernando, J. Sock, T.K. Kim, A review on object pose recovery: from 3D bounding box detectors to full 6D pose estimators, Image Vis. Comput. 96 (2020), 103898, https://doi.org/10.1016/j.imavis.2020.103898.

[11] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, N. Navab, SSD-6D: making rgb-based 3D detection and 6D pose estimation great again, in: IEEE International Conference on Computer Vision, Venice, Italy, 2017, pp. 1521–1529, https://doi.org/10.1109/ICCV.2017.169.

[12] Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: a Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes, arXiv preprint, 2018. https://arxiv.org/abs/1711.00199. v3.

[13] Y.S. He, W. Sun, H.B. Huang, J.R. Liu, H.Q. Fan, J. Sun, PVN3D: a deep point-wise 3D keypoints voting network for 6DoF pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington State, USA, 2020, pp. 11629–11638, https://doi.org/10.1109/CVPR42600.2020.01165.

[14] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Llic, K. Konolige, N. Navab, V. Lepetit, Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes, in: International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 858–865, https://doi.org/10.1109/ICCV.2011.6126326.

[15] N. Pereira, L.A. Alexandre, MaskedFusion: mask-based 6D object pose estimation, in: IEEE International Conference on Machine Learning and Applications, Florida, USA, 2020, pp. 71–78, https://doi.org/10.1109/ICMLA51294.2020.00021.

[16] L. Zou, Z.J. Huang, F.J. Wang, Z.W. Yang, G.P. Wang, CMA: cross-modal attention for 6D object pose estimation, Comput. Graph. 97 (2021) 139–147, https://doi.org/10.1016/j.cag.2021.04.018.

[17] Y. Di, R.D. Zhang, Z.Q. Lou, F. Manhardt, X.Y. Ji, N. Navab, F. Tombari, GPV-pose: category-level object pose estimation via geometry-guided point-wise voting, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Louisiana, USA, 2022, pp. 6771–6781, https://doi.org/10.1109/CVPR52688.2022.00666.

[18] W. Chen, X. Jia, H.J. Chang, J.M. Duan, L.L. Shen, A. Leonardis, FS-Net: fast shape-based network for category-level 6D object pose estimation with decoupled rotation mechanism, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Tennessee, USA, 2021, pp. 1581–1590, https://doi.org/10.1109/CVPR46437.2021.00163.

[19] N.K. Mo, W.S. Gan, N. Yokoya, S.F. Chen, ES6D: a computation efficient and symmetry-aware 6D pose regression framework, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Louisiana, USA, 2022, pp. 6708–6717, https://doi.org/10.1109/CVPR52688.2022.00660.

[20] Y.Z. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, F. Tombari, ZebraPose: coarse to fine surface encoding for 6DoF object pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Louisiana, USA, 2022, pp. 6728–6738, https://doi.org/10.1109/CVPR52688.2022.00662.

[21] D.D. Cai, J. Heikkilä, E. Rahtu, OVE6D: object viewpoint encoding for depth-based 6D object pose estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Louisiana, USA, 2022, pp. 6793–6803, https://doi.org/10.1109/CVPR52688.2022.00668.

[22] Y. You, R.X. Shi, W.M. Wang, C.W. Lu, CPPF: towards robust category-level 9D pose estimation in the wild, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Louisiana, USA, 2022, pp. 6856–6865, https://doi.org/10.1109/CVPR52688.2022.00674.

[23] C.R. Qi, O. Litany, K.M. He, L.J. Guibas, Deep hough voting for 3D object detection in point clouds, in: IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), 2019, pp. 9276–9285, https://doi.org/10.1109/ICCV.2019.00937.

[24] Y. Zhou, O. Tuzel, VoxelNet: end-to-end learning for point cloud based 3D object detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Utah, USA, 2018, pp. 4490–4499, https://doi.org/10.1109/CVPR.2018.00472.

[25] K. Park, T. Patten, M. Vincze, Pix2Pose: pixel-wise coordinate regression of objects for 6D pose estimation, in: IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), pp. 7667–7676, https://doi.org/10.1109/ICCV.2019.00776.

[26] M. Sundermeyer, Z.C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3D orientation learning for 6D object detection from RGB images, in: European Conference on Computer Vision, Munich, Germany, 2018, pp. 712–729, https://doi.org/10.1007/978-3-030-01231-1_43.

[27] M. Sundermeyer, M. Durner, E.Y. Puang, Z.C. Marton, N. Vaskevicius, K.O. Arras, R. Triebel, Multi-path learning for object pose estimation across domains, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington State, USA, 2020, pp. 13913–13915, https://doi.org/10.1109/CVPR42600.2020.01393.

[28] C.R. Qi, H. Su, K.C. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3D classification and segmentation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Hawaii, USA, 2017, pp. 77–85, https://doi.org/10.1109/CVPR.2017.16.

[29] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, arXiv preprint, 2017. https://arxiv.org/abs/1706.02413v1.

[30] G. Gao, M. Lauri, Y.L. Wang, X.L. Hu, J.W. Zhang, S. Frintrop, 6D object pose regression via supervised learning on point clouds, in: IEEE International Conference on Robotics and Automation, Paris, France, 2020, pp. 3643–3649, https://doi.org/10.1109/ICRA40945.2020.9, 197461.

[31] G. Gao, M. Lauri, X.L. Hu, J.W. Zhang, S. Frintrop, CloudAAE: learning 6D object pose regression with on-line data synthesis on point clouds, in: IEEE International Conference on Robotics and Automation, Xi'an, China, 2021, pp. 11081–11087, https://doi.org/10.1109/ICRA48506.2021.9561475.

[32] M. Sundermeyer, Z.C. Marton, M. Durner, R. Triebel, Augmented autoencoders: implicit 3D orientation learning for 6D object detection, Int. J. Comput. Vis. 128 (2020) 714–729, https://doi.org/10.1007/s11263-019-01243-8.

[33] Y.F. Shi, J.W. Huang, X. Xu, Y.F. Zhang, K. Xu, StablePose: learning 6D object poses from geometrically stable patches, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Tennessee, USA, 2021, pp. 15217–15226, https://doi.org/10.1109/CVPR46437.2021.01497.

[34] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, X. Zabulis, T-LESS: an RGB-D dataset for 6D pose estimation of texture-less objects, in: IEEE Winter Conference on Applications of Computer Vision, California, USA, 2017, pp. 880–888, https://doi.org/10.1109/WACV.2017.103.

[35] B. Drost, M. Ulrich, N. Navab, S. Ilic, Model globally, match locally: efficient and robust 3D object recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, California, USA, 2010, pp. 998–1005, https://doi.org/10.1109/CVPR.2010.5540, 108.

[36] S. Hinterstoisser, V. Lepetit, N. Rajkumar, K. Konolige, Going further with point pair features, in: European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 834–848, https://doi.org/10.1007/978-3-319-46487-9_51.
[37] J. Vidal, C.Y. Lin, X. Lladó, R. Martí, A method for 6D pose estimation of free-form rigid objects using point pair features on range data, Sensors 18 (2018) 2678, https://doi.org/10.3390/s18082, 678.
[38] D.P. Li, H.Y. Wang, N. Liu, X.M. Wang, J. Xu, 3D object recognition and pose estimation from point cloud using stably observed point pair feature, IEEE Access 8 (2020) 44335–44345, https://doi.org/10.1109/ACCESS.2020.2978255.
[39] B. Drost, S. Ilic, 3D object detection and localization using multimodal point pair features, in: Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, Zurich, Switzerland, 2012, pp. 9–16, https://doi.org/10.1109/3DIMPVT.2012.53.
[40] J.W. Guo, X.J. Xing, W.Z. Quan, D.M. Yan, Q.Y. Gu, Y. Liu, X.P. Zhang, Efficient center voting for object detection and 6D pose estimation in 3D point cloud, IEEE Trans. Image Process. 30 (2021) 5072–5084, https://doi.org/10.1109/TIP.2021.3078109.
[41] H.W. Deng, T. Birdal, S. Ilic, PPFNet: global context aware local features for robust 3D point matching, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Utah, USA, 2018, pp. 195–205, https://doi.org/10.1109/CVPR.2018.00028.
[42] L. Jiang, H.S. Zhao, S.S. Shi, S. Liu, C.W. Fu, J.Y. Jia, PointGroup: dual-set point grouping for 3D instance segmentation, in: IEEE/CVF conference on computer vision and Pattern recognition, Washington State, USA, 2020, pp. 4866–4875, https://doi.org/10.1109/CVPR42600.2020.00492.
[43] R.Z. Ge, Z.Z. Ding, Y.H. Hu, Y. Wang, S.J. Chen, L. Huang, Y. Li, AFDet: Anchor Free One Stage 3D Object Detection, arXiv preprint, 2020. https://arxiv.org/abs/2006.12671v2.
[44] T.W. Yin, X.Y. Zhou, P. Krahenbuhl, Center-based 3D object detection and tracking, in: IEEE/CVF conference on computer vision and pattern recognition, Tennessee, USA, 2021, pp. 11779–11788, https://doi.org/10.1109/CVPR46437.2021.01161.
[45] Z.K. Dong, S.C. Liu, T. Zhou, H. Cheng, L. Zeng, X.Y. Yu, H.D. Liu, PPR-Net: point-wise pose regression network for instance segmentation and 6D pose estimation in bin-picking scenarios, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Macau, China, 2019, pp. 1773–1780, https://doi.org/10.1109/IROS40897.2019.8967895.
[46] L. Zeng, W.J. Lv, Z.K. Dong, Y.J. Liu, PPR-Net++: accurate 6-D pose estimation in stacked Scenarios, IEEE Trans. Autom. Sci. Eng. 4 (2022) 3139–3151, https://doi.org/10.1109/TASE.2021.3108800.
[47] B. Graham, M. Engelcke, L. van der Maaten, 3D semantic segmentation with submanifold sparse convolutional networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Utah, USA, 2018, pp. 9224–9232, https://doi.org/10.1109/CVPR.2018.00961.
[48] Y. Yan, Y.X. Mao, B. Li, SECOND: sparsely embedded convolutional detection, Sensors 18 (2018) 3337, https://doi.org/10.3390/s18103337.
[49] E.B. Dam, M. Koch, M. Lillholm, Quaternions, Interpolation and Animation, Technical Report, 1998. https://stuff.mit.edu/afs/athena/course/2/2.998/www/QuaternionReport1.pdf.
[50] J. Solà, Quaternion Kinematics For the Error-State Kalman filter, arXiv preprint, 2017. https://arxiv.org/abs/1711.02508v1.
[51] Q.Y. Zhou, J. Park, V. Koltun, Open3D: A Modern Library for 3D Data Processing, arXiv preprint, 2018. https://arxiv.org/abs/1801.09847v1.
[52] K. Kleeberger, C. Landgraf, M.F. Huber, Large-scale 6D object pose estimation dataset for industrial bin-picking, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Macau, China, 2019, pp. 2573–2578, https://doi.org/10.1109/IROS40897.2019.8967594.
[53] R. Brégier, F. Devernay, L. Leyrit, J.L. Crowley, Symmetry aware evaluation of 3D object detection and pose estimation in scenes of many parts in bulk, in: IEEE International Conference on Computer Vision Workshops, Venice, Italy, 2017, pp. 2209–2218, https://doi.org/10.1109/ICCVW.2017.258.
[54] R. Brégier, F. Devernay, L. Leyrit, J.L. Crowley, Defining the pose of any 3D rigid object and an associated distance, Int. J. Comput. Vis. 126 (2018) 571–596, https://doi.org/10.1007/s11263-017-1052-4.
[55] E. Zhang, Y. Zhang, Average precision. Encyclopedia of Database Systems, Springer, Massachusetts, USA, 2009, pp. 192–194, https://doi.org/10.1007/978-0-387-39940-9_482.
[56] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes, in: Asian Conference on Computer Vision, Daejeon, Korea (South), 2012, pp. 548–562, https://doi.org/10.1007/978-3-642-37331-2_42.
[57] K. Kleeberger, M.F. Huber, Single shot 6D object pose estimation, in: IEEE International Conference on Robotics and Automation, Paris, France, 2020, pp. 6239–6245, https://doi.org/10.1109/ICRA40945.2020.9197207.
[58] J. Sock, K.I. Kim, C. Sahin, T.K. Kim, Multi-task Deep Networks for Depth-Based 6D Object Pose and Joint Registration in Crowd Scenarios, arXiv preprint, 2018. https://arxiv.org/abs/1806.03891. v1.
[59] C.G. Zhuang, Z. Wang, H. Ding, Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking, Robot. Comput.-Integr. Manuf. 68 (2021), 102086, https://doi.org/10.1016/j.rcim.2020.102086.
[60] Bin Picking Studio and Vision Controller, Photoneo, https://www.photoneo.com/.