2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

ParametricNet++: A 6DoF Pose Estimation Network with Sparse Keypoint Recovery for Parametric Shapes in Stacked Scenarios

Yi Han Xie[1,†], Wei Jie Lv[2,†], Xin Yu Zhang[1], Yi Hong Chen[1], Long Zeng[1*]

***Abstract*— Most industrial parts are designed from parametric shapes with the properties of diversity and uncertainty. We propose a 6DoF pose estimation network, ParametricNet++, based on pointwise regression and sparse keypoint recovery, which is extended from ParametricNet to include optimizations of keypoint selection and prediction. Keypoint selection optimization selects geometrically unique keypoints and keypoint groups to reduce the difficulty of scene keypoint prediction and template keypoint recovery. Keypoint prediction optimization predicts keypoints from rough to precise, which improves the accuracy of scene keypoint prediction and template keypoint recovery. Compared with other state-of-the-art methods, the average of APs of ParametricNet++ is improved by over 15% on the public Siléane dataset, and the average of mAPs is improved by 12% and 14% on L-dataset and G-dataset from Parametric dataset, respectively. In particular, ParametricNet++ outperforms our original ParametricNet by 5% for both learning and generalization ability evaluation on the Parametric dataset. The experimental results demonstrate that ParametricNet++ lays a solid foundation for robot grasping in industrial scenarios.**

***Index Terms*— 6DoF pose estimation, pointwise regression, keypoint recovery, stacked scenario, robot grasping.**

I. INTRODUCTION

PARAMETRIC techniques are widely used in engineering product design [1]. A parametric shape is a template described by a set of shape parameters and constraints [2], [3], which can instantiate a family of part objects. In a robot-based assembly system, different part objects from the same shape template, e.g., multiple types of nuts, are grasped from different stacked bins to assemble an industrial product. Meanwhile, the parameter values of the part objects are frequently adjusted based on production requirements. Therefore, 6DoF object pose (i.e., 3D translation and 3D rotation) estimation (OPE) for parametric shapes is challenging for such a vision-guided robot grasping system, since parametric shapes have the properties of diversity and uncertainty.

Existing 6DoF OPE methods with the generalization ability for unknown part objects, target at unseen objects without exact 3D models, and are roughly divided into direct regression methods [4], [5], [6], [7], [8], [9] and dense object recovery methods [10], [11], [12]. However, the generalization abilities of direct regression methods are unsatisfactory due to the nonlinear rotation space [13], and dense object recovery methods have difficulty accurately estimating the pose and size of objects with severe occlusion due to the heavy reliance of the predicted dense correspondence. Our group previously proposed a sparse keypoint recovery method, ParametricNet [14], for parametric shapes in stacked scenarios, which has a stronger generalization ability and a more flexible representation.

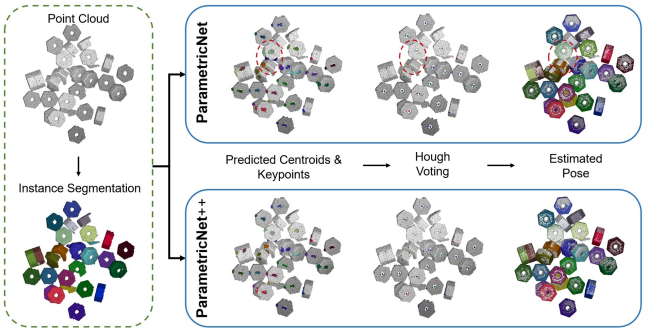


Fig. 1. Comparison of ParametricNet and ParametricNet++: due to the keypoint selection and prediction problems, ParametricNet predicts wrong keypoints and poses for some individual instances shown in the dashed circles, which can be corrected by ParametricNet++.

ParametricNet [14] is a 6DoF pose estimation network for parametric shapes based on pointwise regression and sparse keypoint recovery, which outperformed state-of-the-art methods at that time by a large margin on the benchmark Siléane and Parametric datasets, and had excellent performance at grasping unknown part objects in real-world experiments. The key idea is to map each point from the original point cloud space to the centroid and keypoint spaces to achieve instance segmentation and keypoint prediction. According to the calculated parameter values, the sparse keypoints in the template space are reconstructed to establish the sparse correspondence between the scene and the template, and the 6DoF pose is finally solved by a least-squares fitting algorithm [15].

However, our experiments revealed two major problems. First, keypoint selection is crucial to define the geometrically unique (GU) keypoint labels in the stacked scene and the template keypoint prior information in the template space, which directly affect the difficulty of scene keypoint prediction and template keypoint recovery. The judgment conditions for the GU keypoint are too harsh in ParametricNet, and the keypoint group in the scene may not be GU (see section III-C), which decreases the performance of pose

† Equal contribution.

This research was funded by the Guangdong Natural Science Fund-General Programme (No. 2022A1515011234) and the National Natural Science Foundation of China (No. 61972220) .

1 Department of Advanced Manufacturing, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China {xieyh21, Zhangxy20, chenyiho21}@mails.tsinghua.edu.cn

2 Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China lwj19@mails.tsinghua.edu.cn

* Corresponding author. zenglong@sz.tsinghua.edu.cn

estimation. Second, the design of the keypoint prediction module affects the accuracy of scene keypoint prediction and template keypoint recovery. As the original network structure and loss function in the keypoint prediction module may predict scattered keypoint clusters and fail to learn GU keypoints in the stacked scene (detailed in Section III-D), optimization of the keypoint prediction is needed.

We propose a 6DoF pose estimation network, ParametricNet++, for parametric shapes. It is extended from ParametricNet [14], with keypoint selection and prediction optimizations. For keypoint selection optimization, we define a new judgment condition for the GU keypoint and introduce the charity constraint for the GU keypoint group. For keypoint prediction optimization, we design a new keypoint prediction module architecture based on the idea of prediction from rough to precise, and a new keypoint prediction module loss with rule learning weights. As shown in Fig. 1, due to keypoint selection and prediction problems, ParametricNet may fail to predict the keypoints of some individual instances in the stacked scene, resulting in inferior pose estimation performance, but this can be corrected by ParametricNet++.

We experimentally evaluated the performance of ParametricNet++ on the benchmark Siléane [16] and Parametric [14] datasets. In learning ability, our method outperformed state-of-the-art methods OP-Net [17] and PPR-Net [5] by 15% in the average of APs (average precision) on the Siléane dataset, and by 12% in the average of mAPs (mean average precision) on the Parametric dataset, respectively. In generalization ability, our method outperformed PPR-Net by over 14% in the average of mAPs. Our method outperformed our original ParametricNet [14], by 5% for both learning and generalization ability evaluations on the Parametric dataset.

In summary, we substantially improve our previous work ParametricNet [14] and make the following contributions:

- We propose a 6DoF pose estimation network with keypoint selection and prediction optimizations for parametric shapes, based on pointwise regression and sparse keypoint recovery.
- We propose effective keypoint selection optimization, including threshold-based GU keypoint selection and charity-based GU keypoint group selection.
- We propose keypoint prediction optimization with a new keypoint prediction module.

II. Related Work

In this section, we review the related works on 6DoF OPE methods with the generalization ability for unseen objects, which are roughly divided into direct regression, dense object recovery and sparse keypoiont recovery methods.

A. Direct regression methods

Some works [4], [5], [6], [7] first extract high-dimensional features, and directly regress the 6DoF pose of objects by the network. Xiang et al. [4] extracted features of each object in the scene from RGB images, which were fed into the network to directly regress the location and rotation quaternion. However, features extracted from RGB lacked depth information and were sensitive to illumination, occlusion, and distortion, which could degrade network performance. Dong et al. [5], [6] took the point cloud as input and proposed a pointwise regression network with a Hough voting scheme to directly regress the 6DoF pose. Due to the shape and size variations of objects, Chen et al. [7] achieved object detection with RGB, and proposed a network with an online 3D deformation mechanism for data augmentation and feature extraction of the point cloud, which directly regressed the 6DoF pose and size of the object. However, non-texture and sparsity of the point cloud limited the richness of the extracted features. To improve the generalization of the network for unknown objects, some works [8], [9] have obtained features with more abundant information by fusing different modal information. Wang et al. [8] extracted pixel-wise RGB-D fusion features to directly regress the pose of an object, which was a great improvement over some RGB-based methods. Chen et al. [9] used both geometric and category information to produce dense fusion features to regress a 6DoF pose and calculate the size of objects by reconstruction. Obviously, the performance of these methods is determined by the features extracted. In addition, regressing in a nonlinear pose space (e.g., 3D rotation) is more challenging than in linear space (e.g., 3D translation), where it is difficult to generalize to unknown objects with various shapes.

B. Dense object recovery methods

These methods reconstruct dense 3D coordinates of unknown objects in a normalized space, then establish a dense 3D-3D correspondence to solve the 6DoF pose and size. Wang et al. [10] proposed the normalized object coordinate space (NOCS), in which all instances from a category are reconstructed as the same normalized model. A network directly predicts pixel-wise 3D coordinates of each object in NOCS on RGB images to establish a dense 3D-3D correspondence, and solves the 6DoF pose and size by a Umeyama algorithm [18]. However, the lack of depth information leads to certain limitations. Tian et al. [11] extended NOCS by adding the category shape priors, and considered the texture and depth information to predict the deformation of the object. Wang et al. [12] proposed a cascading network based on NOCS to fully integrate RGB, depth, and shape information of different objects from the same category, and a recurrent network to optimize the reconstruction of objects in NOCS. These methods encounter difficulty in accurately estimating the 6DoF pose and size of objects with little texture and serious occlusion, since they heavily rely on the prediction of dense correspondence by the network.

C. Sparse keypoint recovery methods

Keypoints are a more flexible representation than dense methods. In addition, the above methods can only get the global size of the object, e.g., the size of the bounding box. These methods cannot predict the parameter values of objects with an artificially defined local size, such as parametric shapes with driven parameters in industry. Lv et al. [14] proposed ParametricNet, a 6DoF OPE network

with pointwise regression and sparse keypoint recovery, for the 6DoF pose estimation and parameter value prediction of parametric shapes in stacked scenarios. Our method extends ParametricNet, with keypoint selection and prediction optimizations to further improve the performance of keypoint recovery and pose estimation.

III. Problem Identification

This section revisits the keypoint selection process and the architecture of ParametricNet, and then analyzes its limitations to identify our research problems.

A. Revisiting keypoint selection for ParametricNet

The keypoint selection strategy in ParametricNet is to select template keypoints based on driven parameters and then we transform them to the scene [14]. The scene keypoint kp has an equivalent keypoint set $\Re(kp)$, which has four cases, i.e., spherical-, rotational-, finite-, and non-symmetry [14]. Keypoints with symmetry lack geometric uniqueness, which will cause confusion in the learning phase [14]. Therefore, we select the GU keypoint label in the stacked scene by choosing the keypoint closest to the camera from its equivalent keypoint set. We define three rules to choose the keypoint closest to the camera:

$$R1 : kp_c = \left\{ kp | z_{kp} = \min_{p \in \Re(kp)} z_p \right\} \tag{1}$$

$$R2 : kp_c = \left\{ kp | y_{kp} = \min_{p \in \Re(kp)} y_p \right\} \tag{2}$$

$$R3 : kp_c = \left\{ kp | x_{kp} = \min_{p \in \Re(kp)} x_p \right\}, \tag{3}$$

where $\Re(kp) = \{p_i\}_{i=1}^{n}$ is the equivalent keypoint set, and z_{kp}, y_{kp}, x_{kp} are the z, y, and x coordinates, respectively, of kp in the camera frame.

The GU keypoint selection process is as follows. In $\Re(kp)$, we first select the GU keypoint with the smallest z coordinate as the label according to $R1$. If there are keypoints with equal z coordinates, we redefine $\Re(kp)$ with these keypoints, and select the GU keypoint with the smallest y coordinate according to $R2$. If there are keypoints with equal z and y coordinates, we redefine $\Re(kp)$ with these keypoints, and select the GU keypoint with the smallest x coordinate according to $R3$.

B. Revisiting ParametricNet

As shown in Fig. 2, ParametricNet consists of six modules. PointSIFT is adopted as the backbone to extract the pointwise high-dimensional features of size $N_p \times N_f$ from the point cloud, where N_f is the dimension of the features. Extracted features are applied in centroid, keypoint and visibility predictions. Centroid prediction module M_C regresses pointwise centroids of individual instances to which each point belongs. In the centroid space, we achieve instance segmentation with a clustering algorithm, and pointwise centroids vote for the final predicted centroids of size $d \times 3$, where d is the number of individual instances in the scene. Similarly, keypoint prediction module M_{KP} regresses the pointwise keypoints of size $N_p \times 3m$, which vote for the final predicted keypoints of size $d \times 3m$, where m is the number of keypoints of the parametric shapes.

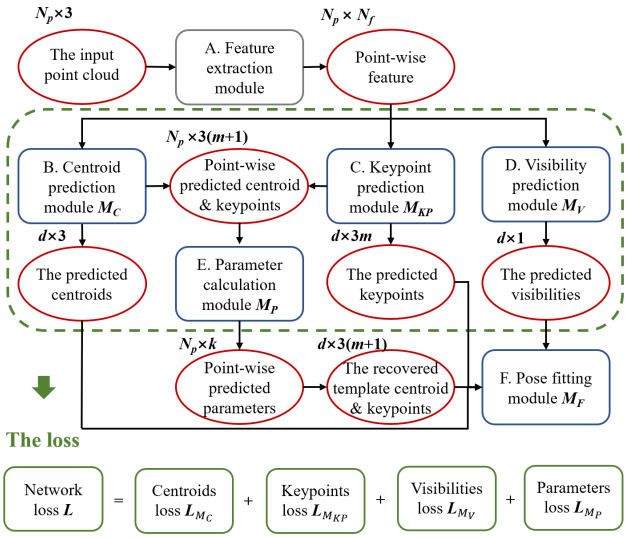


Fig. 2. ParametricNet components and their loss functions.

With the pointwise keypoints and centroid in the scene, the pointwise parameter values of size $N_p \times k$ can be computed according to the template's prior information, where k is the number of the template's parameters. In the template space, the template's keypoints and centroid of size $d \times 3(m+1)$ are reconstructed. In addition, the visibility prediction module M_V helps to filter the ungraspable individual instances with high occlusion rates. Taking the keypoint sparse correspondence between the template and each individual instance in the scene as input, the pose fitting module M_F solves the final pose by a least-squares fitting algorithm [15].

C. Keypoint selection problem

As mentioned in section III-A, GU keypoint selection is to select the keypoint closest to the camera from the equivalent keypoint set, which is defined by three rules. When we judge which rule is suitable for the current GU keypoint selection, the judgment condition is whether the coordinate values are strictly equal, i.e., generating keypoint labels based on exact numerical equality. As shown in Fig. 3, when we select the GU keypoint between the equivalent $kp_3^{(1)}$ and $kp_3^{(2)}$ in the parametric shape TN06, it is obvious that only if the z coordinates of $kp_3^{(1)}$ and $kp_3^{(2)}$ are strictly equal, $R2$ is applied to select $kp_3^{(2)}$ as a GU keypoint label, like the state of c. But it is difficult to judge whether the coordinate values of the equivalent keypoints satisfy strict numerical equality when dealing with irregular point cloud of the stacked scene. Therefore, we need more flexible judgment conditions of the three rules to make GU keypoint prediction less difficult.

Meanwhile, a keypoint group in the scene consisting of a set of GU keypoints may not be geometrically unique

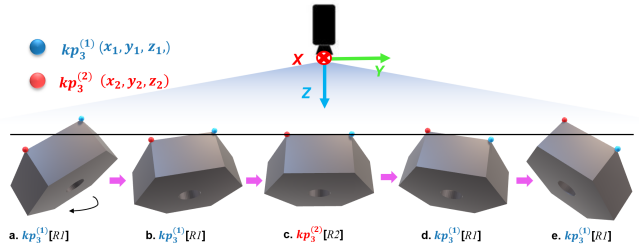


Fig. 3. GU keypoint selection problem.

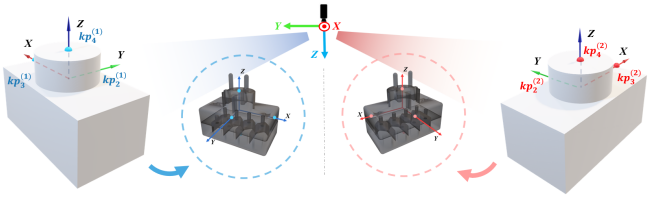
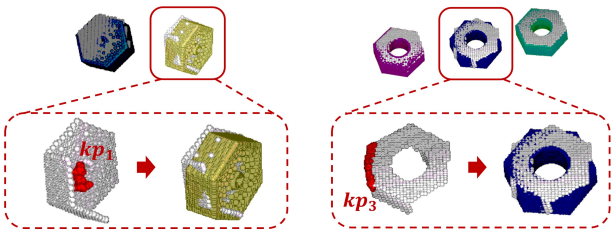


Fig. 4. GU keypoint group selection problem: the model in grey is T-Less20, and the model in white is its converted part object.

in different view perspectives. For the nonparametric object T-Less20, we consider it as a part object from a simple parmetric template, and the blue and red keypoints in Fig. 4 are keypoint groups selected from two view perspectives, belonging to left- and right-handed coordinate systems, respectively. Obviously, these keypoint groups form two chirality structures, which cannot overlap by translation and rotation. In this case, to establish the sparse correspondence between keypoints of the template and the scene, we need to predefine template keypoint groups that satisfy two chirality structures, which undoubtedly increases the difficulty of template keypoint recovery.

D. Keypoint prediction problem

As mentioned in section III-B, the performance of the predicted keypoints in the stacked scene is crucial for pose fitting and the calculation of parameter values to recover template keypoints. As shown in Fig. 5(a), the keypoint prediction module of ParametricNet may regress some deviated and scattered keypoint clusters of kp_1 (center point of upper surface) in TN06, causing deviation error of the final estimated pose. Hence, it is necessary to increase the precision of prediction from the keypoint prediction module.



(a) Scattered keypoint cluster (b) Rule learning imbalance

Fig. 5. Keypoint prediction problem.

In addition, when the network predicts the GU keypoints in the stacked scene, most GU keypoint labels are selected by $R1$. Since GU keypoints suitable for $R2$ and $R3$ are relatively rare, the keypoint prediction module of ParametricNet focuses on learning the GU keypoints suitable for $R1$, and ignores other keypoints, which we define as a rule learning imbalance problem. As shown in Fig. 5(b), since z coordinates of two equivalent keypoints of kp_3 (corner of the hexagon surface) of individual instance in red box, are nearly equal, the keypoint prediction module fails to distinguish which rule should be applied for GU keypoint prediction, which causes the predicted keypoint clusters distributed in long strips, resulting in a wrong estimated pose. Therefore, the loss function of the keypoint prediction module should be adjusted to balance rule learning.

IV. Optimization Methods

We propose ParametricNet++ to solve the keypoint selection and prediction problems for parametric shapes in stacked scenarios. As shown in Fig. 6, it consumes the unordered point cloud directly and estimates the 6DoF pose by five modules. Compared with ParametricNet, ParametricNet++ includes keypoint selection and prediction optimizations, which are discussed in sections IV-A and IV-B, respectively.

A. Keypoint Selection Optimization

To solve the keypoint selection problem in section III-C, keypoint selection optimization optimizes GU keypoint selection and GU keypoint group selection.

To select the GU keypoint from the equivalent keypoint set $\Re(kp)$, we propose the threshold-based GU keypoint selection process shown in Algorithm 1, where $\{z_{p_i}|p_i \in \Re(kp)\}_{i=1}^{n}$, $\{y_{p_i}|p_i \in \Re(kp)\}_{i=1}^{n}$, and $\{x_{p_i}|p_i \in \Re(kp)\}_{i=1}^{n}$ are the z, y, and x coordinates of $\Re(kp)$, respectively. Specifically, we optimize the rule judgment function for the three rules of the keypoint closest to the camera, defined as

$$J(\{k_i\}_{i=1}^{n}, \text{THR}) = \begin{cases} \text{False}, & judgment\ condition \\ \text{True}, & \text{otherwise} \end{cases}, \quad (4)$$

where the $judgment\ condition$ is that $\exists \|k_j - k_l\| < \text{THR}, j \neq l, k_j, k_l \in \{k_i\}_{i=1}^{n}$, $\{k_i\}_{i=1}^{n}$ is a set of coordinate values, and THR is the threshold for judgment. As shown in Algorithm 1, if there are keypoints from the equivalent keypoint set whose coordinates values satisfy the threshold condition of the current rule judgment function, the current rule does not apply to these keypoints, and the process will update $\Re(kp)$ with these keypoints and jump to the rule judgment function of the next rule. As shown in Fig. 7, we select the GU keypoint label from two equivalent keypoints of kp_3, based on the threshold. It is obvious that as long as the z coordinates of $kp_3^{(1)}$ and $kp_3^{(2)}$ enter the threshold range at the same time, the process will turn to $R2$ to select $kp_3^{(2)}$ as the GU keypoint label, like states of b, c, and d. Compared with the original GU keypoint selection process in Fig. 3, our selection process reduces the difficulty of application of $R2$

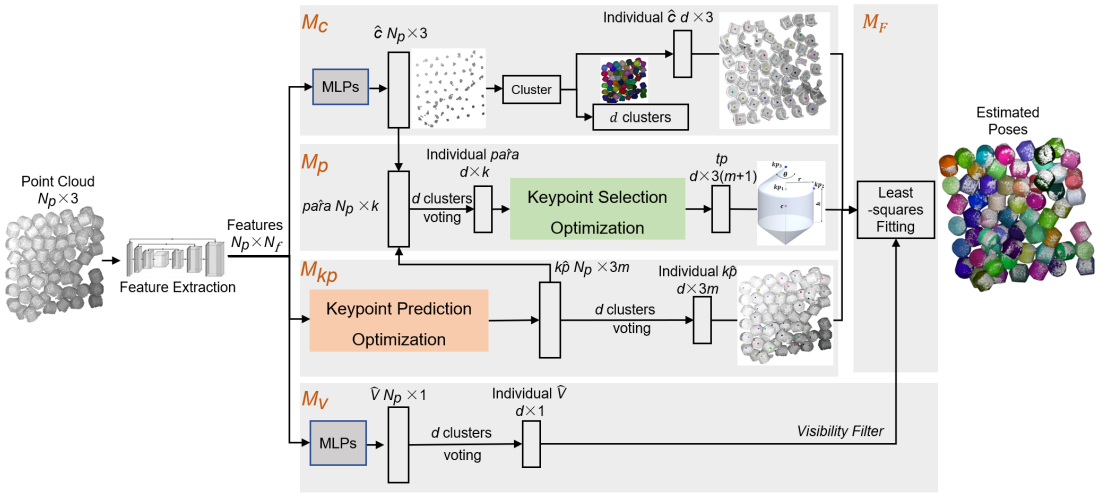


Fig. 6. ParametricNet++ architecture with two new optimizations, i.e., keypoint selection and prediction optimizations, where $\hat{c}$, $\hat{para}$, $\hat{kp}$, and $\hat{V}$ represent predicted centroids, parameters, keypoints, and visibilities, respectively; k is the number of parameters; d is the number of clusters from the instance segmentation to assign the pointwise prediction to d individual instances for voting.

or $R3$ with more lenient judgment conditions, which helps the network to predict the GU keypoints in the stacked scene.

Algorithm 1 Threshold-based GU keypoint selection process

Input: Equivalent keypoint set $\Re(kp)$, threshold THR
Output: Geometrically unique keypoint kp_c

if $J(\{z_{p_i}|p_i \in \Re(kp)\}_{i=1}^{n}$, THR) **then**
  $R1$: $kp_c = \{kp|z_{kp} = \min(z_{p_i}), p \in \Re(kp)\}$
else
  $\Re(kp)' =$ **UPDATE**$(\Re(kp))$
  if $J(\left\{y_{p_i}|p_i \in \Re(kp)'\right\}_{i=1}^{n}$, THR) **then**
    $R2$: $kp_c = \left\{kp|y_{kp} = \min(y_{p_i}), p \in \Re(kp)'\right\}$
  else
    $\Re(kp)'' =$ **UPDATE**$(\Re(kp)')$
    $R3$: $kp_c = \left\{kp|x_{kp} = \min(x_{p_i}), p \in \Re(kp)''\right\}$
  end if
end if

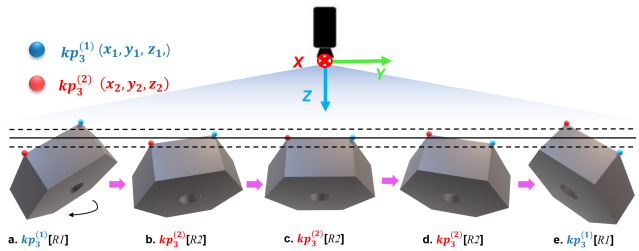


Fig. 7. Example of threshold-based GU keypoint selection process.

To select the GU keypoint group, we employ chirality constraints on the keypoints resulting in the chirality structure. The keypoint with chirality constraints is still geometrically unique, since it is determined by other GU keypoints. As shown in Fig. 8, we select kp_3 and kp_4 by the keypoint

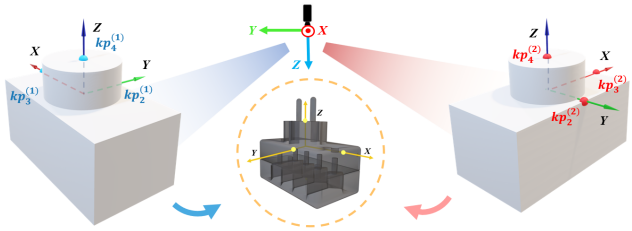


Fig. 8. Example of chirality-based GU keypoint group selection.

closest to the camera from their equivalent keypoint set, and then we select keypoint kp_2 by the right-hand rule based on the relationship of the origin of the local frame, kp_3, and kp_4, which is defined as

$$\mathbf{n}_{kp_2} = \mathbf{n}_{kp_3} \times \mathbf{n}_{kp_4}, \tag{5}$$

where $\mathbf{n}_{kp_2}$, $\mathbf{n}_{kp_3}$, and $\mathbf{n}_{kp_4}$ are the unit vectors pointing to kp_2, kp_3, and kp_4, respectively, from the origin of the local frame. With the chirality constraint, the network only predicts the keypoint group of the left-handed coordinate system, so we simply define one template keypoint group, which reduces the difficulty of template keypoint recovery.

B. Keypoint Prediction Optimization

To address the keypoint prediction problem described in section III-D, we design a keypoint prediction module M_{KP} with a new network architecture and loss function.

As shown in Fig. 9, to solve the problem of deviation and scattered predicted keypoint clusters, the network architecture design of M_{KP} is based on the idea of prediction from rough to precise, with corresponding rough and precise prediction branches. In the rough prediction branch, MLPs are used to regress the pointwise offsets $\left\{\Delta\hat{kp}_i\right\}_{i=1}^{n}$ of size $N_p \times m$ to m keypoints of individual instances to which each point belongs. Then we add it to the point cloud vectors

generated by the element-wise replication of the input point cloud, to obtain the pointwise rough predicted keypoints $\left\{\hat{ckp}_i\right\}_{i=1}^{n}$ of size $N_p \times m$. Different from ParametricNet, in which the pointwise rough predicted keypoints are the only output of its keypoint prediction module, ParametricNet++ has a precise prediction branch, which predicts the pointwise residual vectors $\left\{\hat{\delta kp}_i\right\}_{i=1}^{n}$ of size $N_p \times m$, which represent the three dimension coordinate deviations from m rough predicted keypoints to m precise predicted keypoints in the Euler space. Thus, we can obtain the pointwise precise predicted keypoints $\left\{\hat{kp}_i\right\}_{i=1}^{n}$ of size $N_p \times m$ by adding the pointwise rough predicted keypoints to the residual vectors. As shown in Fig. 10, when predicting the keypoint in the bottom surface of the T-Less20 object, the predicted keypoint clusters of ParametricNet++ are significantly more compact and accurate than those of ParametricNet. The two branches obviously make the predicted keypoint clusters more compact, which can improve the keypoint prediction accuracy of individual instances.

Furthermore, to solve the problem of rule learning imbalance in keypoint prediction, we redesign the loss function of M_{KP} with the rule learning weights for the weighted average. Since we can easily obtain the pointwise rules for each GU keypoint label during the labeling phase, we can construct rule learning weights $\{\omega_i|\omega_i \in [1, L], L > 1\}_{i=1}^{n}$ of size $N_p \times 1$. Specifically, we set the weight values of keypoint labels suitable for $R1$ to 1 which are common in the stacked scene, and set the weight values of other keypoint labels to L, which are relatively rare. The respective loss functions of the rough keypoint prediction branch, precise keypoint prediction branch, and keypoint prediction module M_{KP} are

$$L_{\hat{ckp}_i} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(\left\|kp_i - \hat{ckp}_i\right\|^2 \times \omega_i\right) \tag{6}$$

$$L_{\hat{kp}_i} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(\left\|(kp_i - \hat{kp}_i) - \hat{\delta kp}_i\right\|^2 \times \omega_i\right) \tag{7}$$

$$L_{M_{KP}} = L_{\hat{ckp}_i} + L_{\hat{kp}_i}, \tag{8}$$

where $\hat{ckp}_i$, $\hat{kp}_i$, $\hat{\delta kp}_i$, and kp_i are the pointwise concatenated vectors of rough predicted keypoints, precise predicted keypoints, predicted keypoint residual values, and keypoint labels, respectively ($\hat{ckp}_i$, $\hat{kp}_i$, $\hat{\delta kp}_i$, and $kp_i \in \mathbb{R}^{3m}$). As shown in Fig. 11(a), for the individual instance shown in the red box, ParametricNet fails to predict the GU keypoint because the network fails to learn the relatively rare GU keypoints suitable for $R2$ or $R3$ in the stacked scene. ParametricNet++ increases the learning strength of $R2$ and $R3$ through rule learning weights, and we use threshold-based GU keypoint selection to reduce the application difficulty of keypoint labels suitable for $R2$ and $R3$ in section IV-A. So, as shown in Fig. 11(b), ParametricNet++ predicts GU keypoints suitable for $R2$ or $R3$ without confusion.

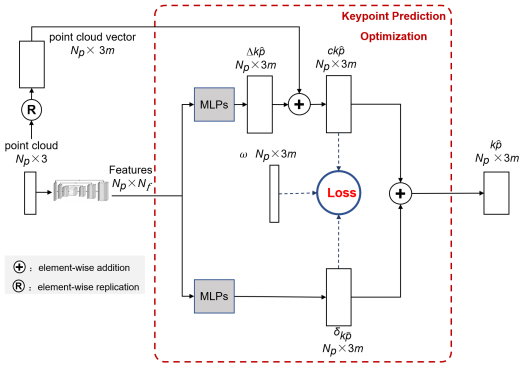
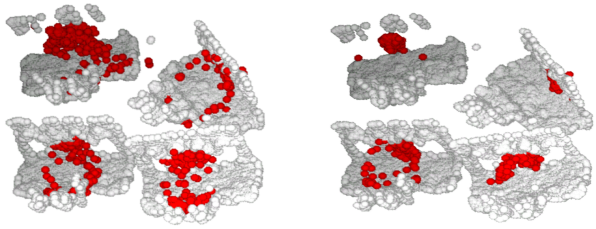


Fig. 9. Architecture of keypoint prediction optimization.



(a) ParametricNet (b) ParametricNet++

Fig. 10. Improvement of predicted keypoint clusters.

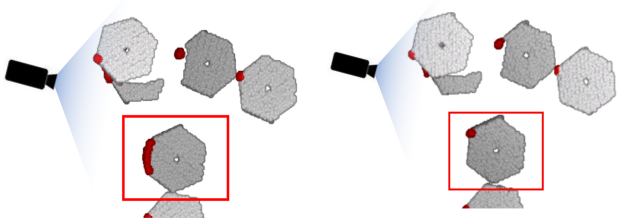
V. Experiments

In this section, we introduce the datasets and the evaluation metrics, and analyze the experimental results.

A. Datasets and evaluation metrics

Siléane dataset. We select four of eight objects in the public Siléane dataset [16]—Candlestick (C.stick), Gear, T-Less20, and T-Less29—to evaluate the learning ability for nonparametric objects. To apply ParametricNet++, we convert these to a part object from a simple shape template consisting of several simple bounding primitives.

Parametric dataset. The Parametric dataset [14] has two testing sets: L-dataset and G-dataset. In the L-dataset, the parameter values of the part objects are the same as those of the training set, and are called known part objects. In the G-dataset, the parameter values of the part objects differ from those of the training set, and are referred to as unknown part objects. We evaluate the learning ability for known



(a) ParametricNet (b) ParametricNet++

Fig. 11. Improvement of rule learning.

part objects on the L-dataset, and evaluate the generalization ability for unknown part objects on the G-dataset.

Evaluation metrics. The pose estimation metric proposed by Brégier [16] is adopted. Individual instances with visibilities greater than 50% are relevant for retrieval. The overall performance of pose estimation is measured by AP, which is the area under the precision-recall curve. For a parametric shape, the evaluation metric adopts mAP, which is the average of the APs of all part objects.

B. Results and comparison

Learning ability evaluation for nonparametric objects on Siléane dataset. To facilitate comparison with other state-of-the-art OPE methods, ParametricNet++ is evaluated on the Siléane dataset, which has unknown noise patterns to mimic real-world data, while being trained on the synthetic data from the Fraunhofer IPA Bin-Picking dataset [19]. As shown in Table I, ParametricNet++ outperforms state-of-the-art methods on the Siléane dataset. Specifically, on the average of APs on the four objects, ParametricNet++ has no post-processing steps in the evaluation process, e.g., duplicate removal or ICP, and outperforms traditional non-learning methods, i.e., PPF [20], [16], PPF PP [20], [16], LINEMOD+ [21], [16], and LINEMOD+ PP [21], [16], by over 58%; outperforms learning-based methods without post-processing steps, i.e., Sock et al. [22], PPR-Net [5], OP-Net Lori_1 [17], and OP-Net Lori_2 [17], by over 19%; and outperforms learning-based methods with post-processing steps, i.e., PPR-Net PP [5], OP-Net Lori_1 PP [17], and OP-Net Lori_2 PP [17], by over 15%. Compared with ParametricNet [14], the performance of ParametricNet++ on the Siléane dataset is further improved, where APs of Candlestick and T-Less20 increase by 1% and 2%, respectively, while APs of Gear and T-Less29 are the same. Based on the above analysis, ParametricNet++ has a strong learning ability for nonparametric objects, as well as a strong domain transfer ability from simulated training data to noisy test data.

TABLE I
EVALUATION ON SILÉANE DATASET.

Object	C.stick	Gear	T-Less20	T-Less29	AVG
PPF	0.16	0.62	0.20	0.19	0.29
PPF PP	0.22	0.63	0.23	0.23	0.33
LINEMOD+	0.38	0.44	0.25	0.20	0.32
LINEMOD+ PP	0.49	0.50	0.31	0.26	0.39
Sock et al.	0.64	–	–	–	–
PPR-Net	0.91	–	0.81	–	–
OP-Net Lori_1	0.94	0.82	0.85	0.51	0.78
OP-Net Lori_2	0.95	0.58	0.56	0.36	0.61
PPR-Net PP	0.95	–	0.85	–	–
OP-Net Lori_1 PP	0.97	0.84	0.88	0.58	0.82
OP-Net Lori_2 PP	0.96	0.60	0.58	0.39	0.63
ParametricNet	0.97	**1.00**	0.92	**0.94**	0.96
ParametricNet++	**0.98**	**1.00**	**0.94**	**0.94**	**0.97**

Learning ability evaluation for known part objects on L-dataset. As shown in Table II, PPR-Net, the winning method of the Object Pose Estimation Challenge for Bin-Picking at IROS 2019 [17], is selected as a baseline. All methods learn all part objects in the training set and are tested by the L-dataset where the parameter values of part objects are the same as those of the training set. It shows that ParametricNet++ can reach 92% on the average of mAPs of four parametric shapes, which is 12% higher than PPR-Net and 5% higher than ParametricNet. Compared with ParametricNet, ParametricNet++ improved by 19% and 3% in the asymmetric TN42 and the finite symmetric TN06, which are difficult to learn. Based on the above analysis, the learning ability of ParametricNet++ on known part objects evaluation improves significantly.

TABLE II
LEARNING ABILITY EVALUATION ON PARAMETRICNET DATASET.

Object	TN06	TN16	TN34	TN42	AVG
PPR-Net	0.80	0.99	**1.00**	0.39	0.80
ParametricNet	0.94	**1.00**	**1.00**	0.52	0.87
ParametricNet++	**0.97**	**1.00**	**1.00**	**0.71**	**0.92**

Generalization ability evaluation for unknown part objects on G-dataset. As shown in Table III, all methods first learn all part objects in the training set and are tested by the G-dataset, where parameter values of part objects are different from those of the training set (by approximately 2%–37% in each parameter value). The average of mAPs of ParametricNet++ in the four parametric shapes is 91%, which is better than that of PPR-Net and ParametricNet by 14% and 5%, respectively. Furthermore, we downsample the number of learned part objects in the training set by 1/3 and 1/5 with the same generalization test set (G-dataset). In 1/3 training mode, ParametricNet++ achieves 88% on the average of mAPs, outperforming PPR-Net and ParametricNet by 15% and 7%, respectively. In 1/5 training mode, ParametricNet++ achieves 77% on the average of mAPs, outperforming PPR-Net and ParametricNet by 18% and 8%, respectively. The results show that when the number of part objects in the training set is gradually reduced, the improvement margin of ParametricNet++ gradually increases.

C. Ablation study

We conduct ablation experiments on optimization performance for ParametricNet++ on the parametric shape TN06, consisting of 128 part objects (half from L-dataset and half from G-dataset). As shown in Table IV, all methods learn all part objects in the training set, where ParametricNet++

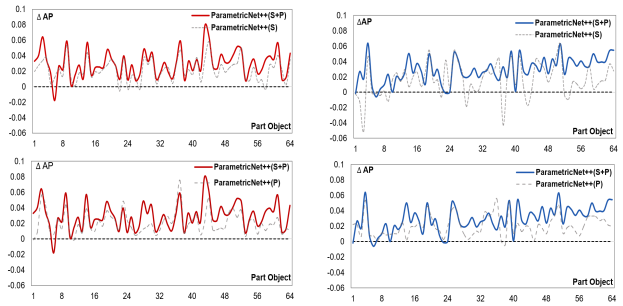
TABLE III
GENERALIZATION ABILITY EVALUATION ON PARAMETRIC DATASET.

Train Mode	Method	TN06	TN16	TN34	TN42	AVG
Learn all	PPR-Net	0.79	0.99	**1.00**	0.28	0.77
	ParametricNet	0.93	**1.00**	**1.00**	0.51	0.86
	ParametricNet++	**0.96**	**1.00**	**1.00**	**0.69**	**0.91**
Learn 1/3	PPR-Net	0.78	0.94	0.96	0.22	0.73
	ParametricNet	0.86	**0.98**	**1.00**	0.41	0.81
	ParametricNet++	**0.94**	**0.98**	**1.00**	**0.60**	**0.88**
Learn 1/5	PPR-Net	0.77	0.56	0.83	0.18	0.59
	ParametricNet	0.86	0.63	0.86	0.39	0.69
	ParametricNet++	**0.92**	**0.71**	**0.89**	**0.57**	**0.77**

(S) only performs keypoint selection optimization, ParametricNet++ (P) only performs keypoint prediction optimization, and ParametricNet++ (S+P) performs both keypoint selection and prediction optimizations. Compared with ParametricNet, ParametricNet++ (S) and ParametricNet++ (P) both improves significantly on mAP, and their combined performance is better than that of a single optimization.

TABLE IV
ABLATION STUDY.

Method	Keypoint Selection	Keypoint Prediction	L-dataset	G-dataset
ParametricNet [14]	×	×	0.939	0.934
ParametricNet++ (S)	✓	×	0.961	0.950
ParametricNet++ (P)	×	✓	0.960	0.954
ParametricNet++ (S+P)	✓	✓	**0.969**	**0.963**



(a) L-dataset (b) G-dataset

Fig. 12. AP improvement curves.

We also evaluate the AP of each part object from the L-dataset and G-dataset. Fig. 12 shows the improved AP of each part object compared with ParametricNet. It can be seen that ParametricNet++ (S) and ParametricNet++ (P) shows improvements on most of the part objects from the L-dataset and G-dataset. The AP improvement curve of ParametricNet++ (S+P) is more stable than that of ParametricNet++ (S) or ParametricNet++ (P), where the maximum increase of a single part object of the L-dataset and G-dataset achieves 8% and 6%, respectively, on AP. The experimental results indicate that our optimizations gained a better learning ability for known part objects and generalization ability for unknown part objects than ParametricNet.

VI. CONCLUSIONS

In this paper, we proposed a 6DoF pose estimation network, ParametricNet++, for parametric shapes based on pointwise regression and sparse keypoint recovery. In ParametricNet++, keypoint selection and prediction optimizations were proposed to improve the pose estimation performance. Experimental results demonstrated the outstanding performance of ParametricNet++ compared with state-of-the-art 6DoF OPE methods. We will release the code soon (`https://github.com/lvwj19/ParametricNet`).

REFERENCES

[1] L. Zeng, Z.-k. Dong, J.-y. Yu, J. Hong, and H.-y. Wang, "Sketch-based retrieval and instantiation of parametric parts," *Computer-Aided Design*, vol. 113, pp. 82–95, 2019.

[2] V. Shapiro and D. L. Vossler, "What is a parametric family of solids?" in *Symposium on Solid Modeling and Applications*, 1995, pp. 43–54.

[3] C. M. Hoffmann and K. J. Kim, "Towards valid parametric CAD models," *Computer-Aided Design*, vol. 33, no. 1, pp. 81–90, 2001.

[4] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

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

[6] L. Zeng, W. J. Lv, Z. K. Dong, and Y. J. Liu, "Ppr-net++: Accurate 6-d pose estimation in stacked scenarios," *IEEE Transactions on Automation Science and Engineering*, 2021.

[7] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis, "Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 1581–1590.

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

[9] D. Chen, J. Li, Z. Wang, and K. Xu, "Learning canonical shape space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 973–11 982.

[10] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[11] M. Tian, M. H. Ang, and G. H. Lee, "Shape prior deformation for categorical 6d object pose and size estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 530–546.

[12] J. Wang, K. Chen, and Q. Dou, "Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4807–4814.

[13] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *IEEE/CVF Conference on CVPR*, 2019, pp. 4556–4565.

[14] L. Zeng, W. J. Lv, X. Y. Zhang, and Y. J. Liu, "Parametricnet: 6dof pose estimation network for parametric shapes in stacked scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 772–778.

[15] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[16] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Symmetry aware evaluation of 3D object detection and pose estimation in scenes of many parts in bulk," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 2209–2218.

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

[18] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[19] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6D object pose estimation dataset for industrial bin-picking," in *2019 IEEE/RSJ IROS*, 2019, pp. 2573–2578.

[20] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE Computer Society Conference on CVPR*, 2010, pp. 998–1005.

[21] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[22] J. Sock, K. I. Kim, C. Sahin, and T. Kim, "Multi-task deep networks for depth-based 6d object pose and joint registration in crowd scenarios." BMVC, 2018.