

Object Pose Estimation for Robotic Grasping based on Multi-view Keypoint Detection

Zheyuan Hu*, Renluan Hou[†], Jianwei Niu^{*‡}, Xiaolong Yu[†], Tao Ren[†], Qingfeng Li[†]

^{*}State Key Laboratory of Virtual Reality Technology and Systems,

School of Computer Science and Engineering, Beihang University, Beijing 100191, China

[†]Hangzhou Innovation Institute, Beihang University, Hangzhou 310051, China

[‡]Zhengzhou University Research Institute of Industrial Technology,

School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China

Email:{huzheyuan18@buaa.edu.cn, jessierlhou@zju.edu.cn, niujianwei@buaa.edu.cn,

yuxiaolong@buaa.edu.cn, taotao_1982@126.com, liqingfengzzu@163.com}

Abstract—Industrial robots can replace human labour to perform a variety of tasks. Among these tasks, robotic grasping is the most primary industrial robot operation. However, conventional robotic grasping methods could become inapplicable for cluttered and occluded objects. To address the issue, we adopt object pose estimation (OPE) to facilitate robotic grasping of cluttered and occluded objects and propose an object detection model based on 2D-RGB multi-view features. The proposed model is built by adding four transpose convolution layers into the Resnet backbone to obtain desirable 2D feature maps of object keypoints in each image. In addition, we design a feature-fusion model to produce 3D coordinates of keypoints from 2D multi-view features based on the volumetric aggregation method, along with a keypoint-detection confidence of each view to assist the optimality judgment of the robotic grasping. Extensive experiments are conducted to verify the accuracy of OPE, and the experimental results indicate the substantial performance improvements of the proposed approach over conventional methods in various scenarios.

Index Terms—Robotic grasping, Keypoint detection, Pose estimation, Multi-view

I. INTRODUCTION

With the significant advances of computing power [1] [2], big data [3] [4] [5], and automation technologies [6] [7], industrial robots have been applied in various areas, such as automobile assembly [8], parcel sorting, and tele-health. Among these tasks, grasping manipulation is the most primary industrial robot operation, which should be designed and implemented with high accuracy and stability. Thanks to the technological advances of automation and information, vision-based robotic grasping has been proposed as an promising paradigm. To perform once satisfactory vision-based grasping, the following three steps should be well designed: Object Pose Estimation (OPE), robot motion planning and grasping operation, among which the OPE is the most important.

Currently, popular industrial OPE methods take 2D image to obtain visual features of the object [9], by simplifying the pose estimation to the prediction of 2D positions and rotation

angles. The robot needs to move the gripper to the appropriate position at preset grasping heights to perform grasping operation. However, 2D visual features are not sufficient to meet the requirements of robotic grasping in 3D spaces, which results in the trend of combining OPE with 3D machine vision. There are two major types of 3D machine vision based OPE: 6-Degree-of-Freedom (6DoF) based pose estimation and 3D keypoint detection based pose estimation.

3D keypoint detection based pose estimation is superior to 6DoF based pose estimation due to the reason that the 3D keypoints can effectively achieve the refined description of the object instead of the object appearance which could be affected by the rotation, scaling, and translation. Therefore, 3D keypoint detection based pose estimation can estimate the pose of the object with satisfactory generalization performance when facing an unknown object with different materials, colors, and sizes. Despite of these advantages, most existing methods need to use monocular color images with depth information as the input of the detection model, which could face troubles in grasping reflective and transparent objects and complex grasping scenes (e.g., overlap and occlusion between objects).

To address the challenge, we propose an object pose estimation method based on multi-view keypoint detection. First, we use the monocular camera placed at the end of the robot to collect RGB pictures and corresponding camera poses from different views during robot motion. Then, we design a multi-view keypoint detection algorithm to calculate the 3D keypoint coordinates and poses of the object. In addition, we design a view-confidence to assist choosing of the most desirable initial grasping pose from different views to avoid obstacle collision.

Finally, we train the model in simulation environments and fine-tune it in real scenarios, which could results in the following benefits: alleviating the high cost of manually labeled data, narrowing the gap between simulation and reality, and enhancing the generalization capability of the detection model. Furthermore, we evaluate the proposed approach by deploying it on a real robot (Rokae XB4) and demonstrate its utility in practical manipulation tasks. Main contributions of this paper are summarized as follows:

Zheyuan Hu and Renluan Hou contributed equally to this paper. Corresponding author: Jianwei Niu and Tao Ren. This work was supported by National Key R&D Program of China (2017YFB1301100).

- We build the robotic grasping dataset, which includes labeled 3D object keypoints of multiple views with each view consists of a large amount of images.
- We propose a multi-view based object keypoint detection model, which can fuse visual features of multiple views to effectively achieve accurate pose estimation of reflective objects, transparent objects, or cluttered and occluded scenarios.
- Extensive experiments are conducted and the experimental results demonstrate the promising accuracy improvements of our approach over the state-of-the-art methods in the baseline datasets (YCB-Video and transparent object dataset).

The paper is organized as follows. In Section II related work is briefly discussed. Section III describes the proposed model in detail. In Section IV, the performance of the model is evaluated via extensive experiments, followed by the conclusion in Section V.

II. RELATED WORK

A. 6DoF object pose estimation

6DoF object pose estimation is a technique to determine the position and pose of an object in a grasping scene, where the position information is mainly represented as a 3D coordinate, and the pose can be represented as pose transformation matrix, Euler angles, rotation quaternions and axis angles, etc. It is widely used in robot grasping, human-computer interaction, and intelligent driving.

6DoF pose estimation can be divided into correspondence-based methods [10], templat-based methods [11], and voting-based methods [12]. The aim of the correspondence-based method is to find the correspondence between the current input data and the existing complete 3D object model. The main idea of templat-based methods is to find the most similar template to the current data from a library of templates labeled with 6DoF object pose. The voting-based methods means that each pixel of the object or each point in the 3D point cloud is used to estimate the 6DoF pose by voting, and each local information in the input data can predict the overall object pose. Such a feature makes this method effective in coping with the problem of objects being partially occluded.

In order to calculate the object's pose, the methods above usually require the known model of the object or the complete point cloud information to establish the correspondence with the current input data, which may lead to the above detection models fail when the 3D information of the object is unknown, and the generalization of the pose estimation is poor.

B. Keypoint detection based object pose estimation

Keypoint detection techniques are mainly used in human pose estimation, face recognition and gesture recognition. For the robot grasping task, the algorithm needs to know the 3D spatial coordinates of the object under the camera coordinate system, and then transform the coordinates into the robot operation space, so it is necessary to research on the 3D keypoint detection. The current approach utilizes 3D

convolutional neural networks for 3D keypoint estimation in real-time from a single depth image [13]. In contrast, multi-view based 3D keypoint detection methods have also been proposed. By feeding pictures from multiple camera views, the detection algorithm restores the 3D information of the detection object based on the positional geometric relationship between different views, and this method is highly resistant to occlusion and has higher detection accuracy.

Based on RGB-depth pictures, [14], [15] use semantic 3D keypoint to represent the object instances of the same category, the detection model is robust to changes in the shape or topology of objects within the category because the keypoint representation ignores geometric details that are not relevant to the grasping task. The 3D keypoints are input into the collision optimization-based robot kinematics solver module, and then object grasping and placement are accomplished using a grasping algorithm that is independent of the target object.

It can be seen that the current 3D keypoint detection methods mainly use color and depth images as the input of the algorithm, but complex grasping scenarios (such as overlapping and occlusion between objects) may affect the detection accuracy of keypoints. In this paper, we consider the multi-view based keypoint detection methods to solve the problem of transparent object detection in occlusion scenarios.

III. MULTI-VIEW KEYPOINT DETECTION BASED OBJECT POSE ESTIMATION

This section proposes a deep learning-based Multi-View Pose Detection algorithm (Multi-View Pose Network, MVP Net), which uses convolutional neural networks (CNN) to extract and fuse the image features of multiple views to estimate the object's pose.

A. Object keypoint detection and pose estimation pipeline

In Fig. 1, the algorithm uses a monocular camera placed at the end of the robot to acquire color images and corresponding camera poses from different views during robot motion. It obtains the bounding box of the grasped object through the object detection module, and crops and scales the original images to improve the detection accuracy. The cropped image of each view is fed into the 2D keypoint detection network to get the corresponding 2D keypoint feature and the keypoint detection confidence, and then the 3D keypoint coordinates are predicted by fusing the keypoint features of different views based on the 3D convolutional network. After that, the object pose is derived based on the 3D spatial position relationship between the keypoints, which lays the foundation for the robot's grasping operation.

The input of the whole detection model is the color image with camera pose in each view, and the output is the 3D coordinates in the camera coordinate system of the first view. Compared with some existing methods for object pose estimation based on image and depth information, this algorithm can achieve effective recognition of occluded, weakly textured, reflective and transparent objects by using the geometric

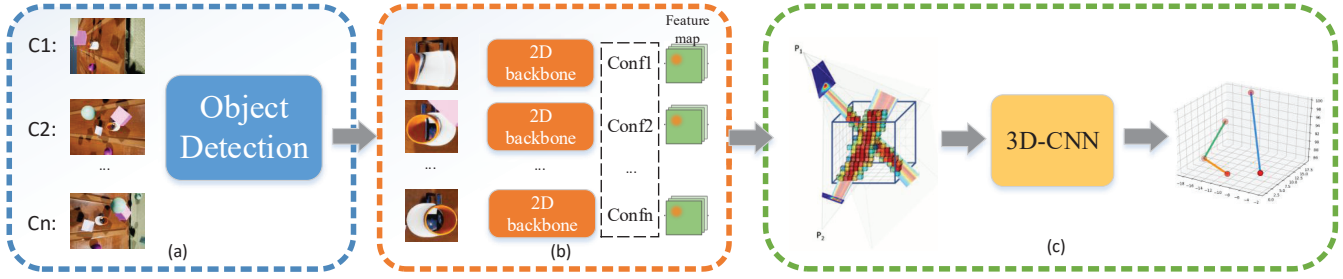


Fig. 1. Object keypoint detection and pose estimation pipeline.

relationship of multiple views. Meanwhile, under the premise of guaranteeing the detection accuracy, this paper lightens the design of the key point detection network, and achieves the feature extraction for all view images in parallel with low latency.

B. Multi-view based object grasping dataset organization

In order to build the dataset of the grasped objects, RGB images with different views and poses need to be captured and labeled with the corresponding 3D key points. We preferred to select the points that represent the local geometric features of the object as keypoints (e.g., center point or boundary points). Also, make the distance of these points as far as possible to better describe the overall information of the object. The multi-view 3D keypoints dataset constructed in this paper contains a total of 99,840 samples.

To enrich the training data and reduce the cost of manual annotated data, we mainly uses simulation data for training. The approximate shape, material and color of the object model are captured randomly through a method similar to domain randomization [16]. The illumination intensity, color and direction of the light source is also randomized; In this paper, NDDS [17] based on Unreal 4 engine is used to generate randomized simulation data.

The real data is used to fine-tune the network after the simulation data training, and improve the applicability of the neural network model in the real scene. After collecting several groups of images from different camera pose, the pixel coordinates of 2D keypoints of each image need to be labeled manually. In this paper, Labelme [44] was used to label the keypoints. Once the pixel coordinates of 2D keypoints $t = [u_n, v_n]^T$ and the corresponding camera pose $P_n = [R_n, T_n] \in \mathbb{R}^{3 \times 4}$ of the pictures under $n (n \geq 2, n \in \{1, 2, \dots, N\})$ view are known, direct linear transformation (DLT) was used to restore 2D key points information of n views to 3D keypoints $W = [x, y, z, 1]^T$, which can be expressed as:

$$\begin{cases} \lambda_n u_n = p_{n,1}w = [R_{n,1}|T_{n,1}][x, y, z, 1]^T \\ \lambda_n v_n = p_{n,2}w = [R_{n,2}|T_{n,2}][x, y, z, 1]^T \\ \lambda_n = p_{n,3}w = [R_{n,3}|T_{n,3}][x, y, z, 1]^T \end{cases}, \quad (1)$$

where λ_n is the depth of the 3D coordinate, which is an unknown variable. After the elimination and matrix transfor-

mation, the 3D coordinates are determined by matrix singular value decomposition (SVD).

C. 2D keypoint detection in a single view

We choose RFB-Net [18] to detect the location of the grasped object in the image at each view and cropped the original image to improve the accuracy of keypoint detection. Then the cropped images are fed into our proposed 2D keypoint detection model.

In Fig. 2, the backbone network of the model uses the ResNet-50, which can effectively avoid the information loss in the process of neural network feature extraction compared with the traditional convolutional network. After the residual network, two different neural networks is used to process the image features, where the confidence network in the upper right part of Fig. 2 is used to predict the confidence $\omega_{n,k}$ of keypoints in each view, as shown in Eq.(2), whose average value represents the confidence $conf_n$ of detection in that view:

$$conf_n = \frac{avg \omega_{n,k}}{K}. \quad (2)$$

The detection confidence of different views can be used as grasping a priori knowledge to assist in designing more effective robot grasping operations, i.e., the higher the confidence of a certain view, the less the object in that view is obscured, the higher the keypoint detection accuracy, and the more favorable the pose presented by the object in that view is for robot grasping. At the same time, this information is also used as the contribution weight of each view to predict the final 3D keypoint coordinates to achieve reasonable usage of object information in different views; the lower right part of Fig. 2 shows the 2D keypoint detection branch. we introduce four deconvolution layers to predict the feature map containing k channels, which corresponds to the k keypoints of the object. Compared with some existing keypoint detection networks that use up-sampling and cross-layer connection, this paper improves the resolution of the feature map in a more concise way, which speeds up the prediction speed of the network model and improves the detection accuracy.

D. Multi-view Feature Fusion Based on 3D CNN

In this paper, the feature map $F_n, n \in \{1, 2, \dots, N\}$ obtained from a 2D keypoint detection network with multiple views

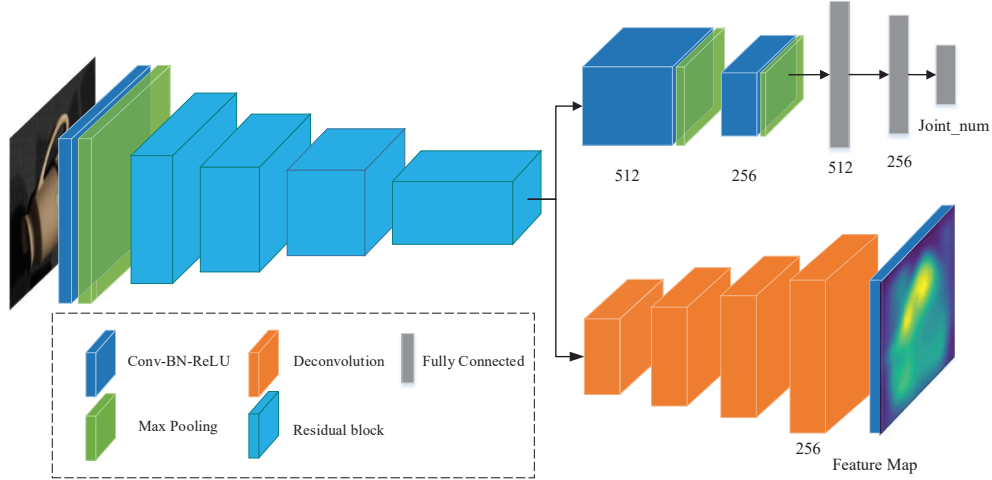


Fig. 2. 2D keypoint detection based on CNN.

is processed using the unprojection. the positive integers N_x, N_y, N_z denote the number of unit cells in x, y , and z directions, a 3D voxel containing $N_x \times N_y \times N_z$ cells is constructed with the object as the center, where the side length of each cell is denoted as L_{size} and the coordinates of the point inside the voxel can be denoted as $i_x, i_y, i_z, i_x, i_y, i_z \in N_x \times N_y \times N_z \times L_{size}$. The input of the unprojection is the set $\{F_n, P_{C_n} | n \in 1, 2, \dots, N\}$ consisting of the feature map under each view with its corresponding camera parameters, and the output is the reconstructed 3D feature map F_{3D} with size $N_x \times N_y \times N_z \times M$, M denotes the number of channels contained in the feature map F_n , and its number is equal to the number of channels of the feature map in each view. The 3D feature map F_{3D} of the k -th key point of the detected object can be calculated as:

$$F_{3D}(i_x, i_y, i_z)_k = \frac{\sum_N (conf_n \cdot F_n\{P_{C_n}(i_x, i_y, i_z)\}_k)}{\sum_N conf_n}. \quad (3)$$

where P_{C_n} denotes the 2D feature map obtained by projection operation on points (i_x, i_y, i_z) in space at the pose of camera C_n , and $F_n\{\cdot\}_k$ denotes the bilinear interpolation mapping of the 2D feature map to obtain the feature map of k channels, which is fused as weights to the 3D feature map of k channels according to the confidence $conf_n$ obtained from the confidence network.

After unprojection, feature extraction is needed for the fused multi-view feature map F_{3D} . Since F_{3D} is a feature representation of 3D spatial information, we use a 3D convolutional network to perform convolutional operations. In Fig. 3, this paper uses a 3D convolutional structure similar to V2V-PoseNet [19], and the network structure of the original V2V-PoseNet is improved for the features of F_{3D} to make it more suitable for the keypoint detection task in this paper.

The input features are encoded and decoded to output the 3D feature map of the k -th keypoint, denoted as V_k . In this

paper, we use softmax to normalize the detection probability of all object keypoints, denoted as V_k' , as follows:

$$V_k' = \frac{\exp(V_k)}{\sum_{i_x=1}^W \sum_{i_y=1}^H \sum_{i_z=1}^D \exp(V_k(i_k))}, \quad (4)$$

where W, H, D denote the width, height and channel depth of the 3D feature map respectively, i_k denotes the voxel coordinates of the k -th key point, and $\exp(\cdot)$ denotes the exponential operation. Finally, the 3D spatial coordinates of the k -th key point of the object is:

$$d_k = \sum_{i_x=1}^W \sum_{i_y=1}^H \sum_{i_z=1}^D i_k \cdot V_k'(i_k), \quad (5)$$

The entire keypoint detection model is trained end-to-end, i.e., inputting 2D image and camera pose information for each view, and outputting 2D keypoint, confidence, and 3D coordinate of the object. Therefore, the loss function of the detection model is divided into two parts:

1) *Loss function for 2D keypoints of all views :*

$$\mathcal{L}_{2d} = \frac{1}{K} \sum_{n \in N} \sum_{k \in K} \|F_{n,k} - F_{n,k}^{gt}\|^2, \quad (6)$$

$F_{n,k}, F_{n,k}^{gt}$ denote the predicted and ground truth of the feature maps output by the 2D keypoint detection network, respectively. The loss function $\mathcal{L}_{2d}$ ensures that the 2D keypoint network can extract the useful keypoint features for each view and accurately predict the keypoint coordinate.

2) *Loss function of 3D keypoints:* $\mathcal{L}_{3d}$ contains two items:

1) the L1 loss of each key point, which is used to predict the 3D spatial information of the keypoints, d_k, d_k^{gt} represent the predicted coordinates and ground truth of each 3D keypoint respectively; 2) the correction item $\log(V_k(d_k^{gt}))$ is used to enhance the accuracy of the featuremap output from the 3D

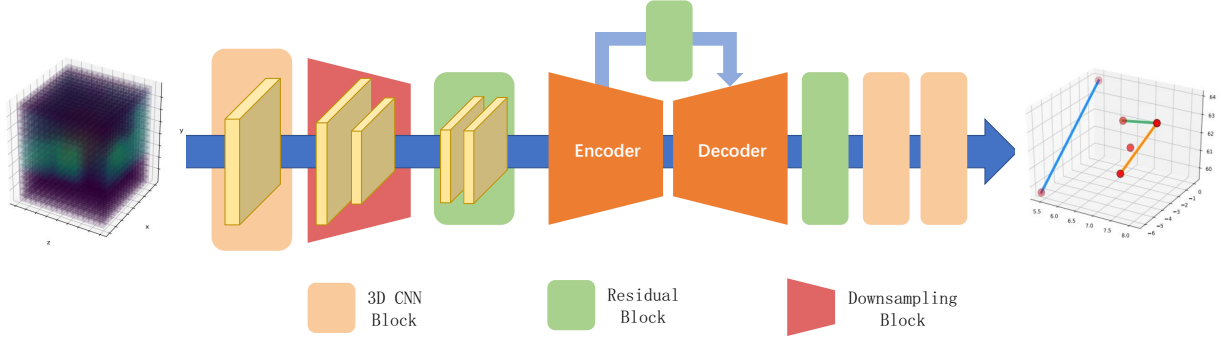


Fig. 3. 3D CNN-based object keypoint detection.

convolutional network, γ is the weight of this item, we set $\gamma = 0.01$.

$$\mathcal{L}_{3d} = \sum_{k \in K} |d_k - d_k^{gt}| - \gamma \log(V_k(d_k^{gt})). \quad (7)$$

In summary, the total loss is the weighted sum of two terms, where the weight $\alpha = 0.8$.

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{2d} + \mathcal{L}_{3d}. \quad (8)$$

By default, four camera views are used for training and testing. In the upprojection stage, the featuremaps of different views are filled into a 3D voxel containing $64 \times 64 \times 64$ unit cells (i.e., $24\text{cm} \times 24\text{cm} \times 24\text{cm}$) according to the pose parameters of each view. The entire detection model is trained using the Adam optimizer, with the initial learning rate set to 0.001 and a decay strategy with a decay factor of 0.3 every 20 training rounds. The model was trained with 200 epochs, and the batch size of training data was 8. The neural network model was implemented in PyTorch 1.1 and trained and tested on an NVIDIA Tesla V100 GPU.

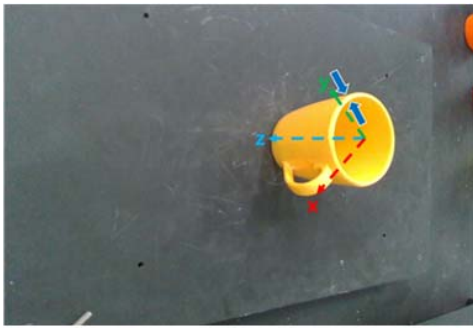


Fig. 4. Determining the pose from 3D keypoints.

After getting the 3D keypoints' coordinates, this paper further determine the pose of the object according to the spatial geometric relationship between different keypoints, which is convenient for the grasping operation of the robot. In Fig. 4, we establishes the coordinate system of the object, taking the mug as an example, and takes the top of the mug as the origin of the coordinate system, calculates the vector from the top to

the bottom of the mug and takes it as the z-axis, the normal vector of the plane formed by the z-axis and the keypoint of the handle of the mug as the y-axis, and then determines the direction of the x-axis according to the nature of three-axis orthogonality.

IV. EXPERIMENTS

A. Detection results in our dataset

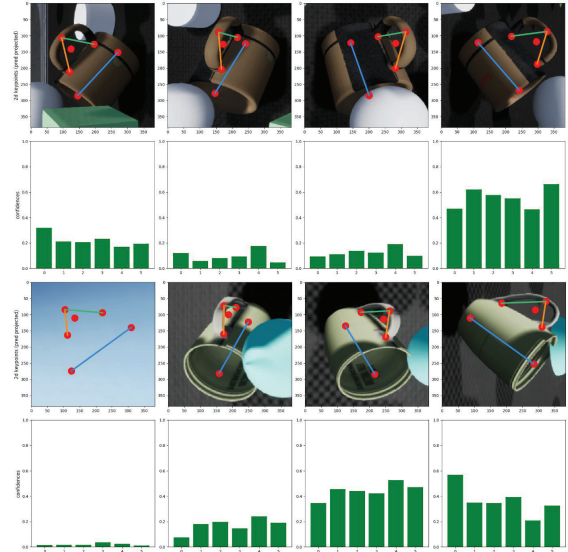


Fig. 5. Object pose estimation and confidence of keypoint detection for different views.

As shown in Fig 5, we first test the performance of the keypoint detection model on the dataset in this paper. With four different camera views and objects of the “mug” category, all 96,000 data items (24,000 samples) of the generated simulation dataset are divided into a training set of 92,080 data items (23,020 samples) and a test set of 3920 data items (980 samples). The size, pose information, and environmental of the objects in each samples are randomly varied. The test set does not overlap with the training set and the unseen models in the training set are added to the test set. In the test set,

we calculate the Mean Absolute Error (MAE) to compare the difference between the prediction and ground truth, which is calculated as follows:

$$MAE = \frac{1}{K} \sum_{k \in K} |d_k - d_k^{gt}|. \quad (9)$$

In Table I, Using the keypoint detection model trained in the simulation data directly to the real data, the detection results of the real data are still poor compared to the results in the simulation data set even after using domain randomization such as object shape, material and ambient lighting. This also indicates that the data generated by domain randomization still cannot be exactly similar to the real data distribution, e.g., the data images collected in the real environment may have noise or blurred images due to robotic arm motion, which are not taken into account in the simulated data. Therefore, in order to improve the detection accuracy on real data and reduce the training cost, we fine-tune the neural network parameters using real data on the basis of the detection model trained on simulated data. The experimental results show that after fine-tuning, the MAE is significantly reduced, and the average MAE is 9.31mm, and the samples with less than 2cm detection threshold account for 97.8% of the total test samples, meeting the need for real object grasping.

B. Detection results in the TOD dataset

The Transparent Object Dataset (TOD) [20] is a dataset containing 15 transparent objects in 6 classes, and the whole dataset contains a total of about 48,000 images, with each object instance sample containing color images of both two camera views, depth images of the left view, and manually labeled 3D key points of the object.

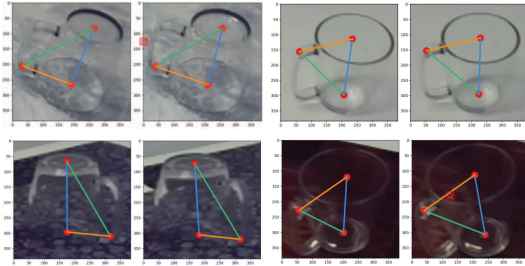


Fig. 6. Experimental results in different background environments.

In Table II, we compare the object detection results of our method with DenseFusion [21] and Keypose [20] on the TOD dataset, where DenseFusion is a 6Dof pose estimation method and Keypose is a binocular keypoint detection method. DenseFusion determines the object pose by fusing color and depth features, and when facing transparent objects, using the depth information captured by the camera will result in huge errors. In contrast, Keypose and our method are both based on multi-view detection, and the detection result of both methods is significantly better than DenseFusion, which shows that multi-view information is very effective for the detection of transparent objects. However, in Fig 6, our method still has

a significant advantage over Keypose due to the inclusion of multiple deconvolution layers, which ensures that the neural network contains rich semantic features without losing the geometric features of the key points. In addition, the existence of the confidence level of key points in each view, the 3D key point information is predicted by weighted fusion between different views, which is more flexible than the direct fusion of the two views of the Keypose, so the detection accuracy is higher.

C. Detection results in the YCB-Video dataset

Table III shows the comparison results of our method with three popular 6Dof pose estimation methods, PointFusion [22], PoseCNN [11] and DenseFusion [21] on the YCB-Video dataset [11], and it can be seen that our method outperforms the other three methods at the overall level. It is worth noting that all three methods use "color-depth" information as the input to the detection model, they need to match the local depth information of the object with the point cloud information of the known object when the object is occluded.

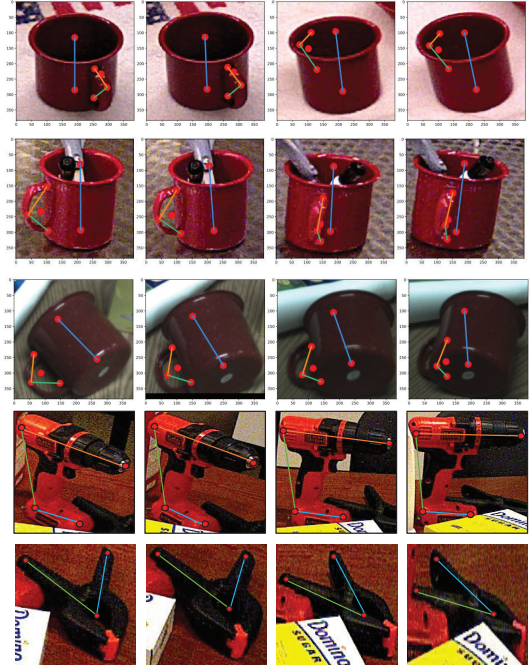


Fig. 7. Pose estimation for different categories in YCB-Video dataset.

It can also be seen from Table III that the PoseCNN and DenseFusion methods perform well for some objects, such as mugs and scissors, but the pose accuracy obtained by iterative regression for these methods is poor when facing symmetric objects like clamps, because the iterative closest point (ICP) algorithm may regress to get multiple solutions. The ICP algorithm also takes a lot of time for iterative computation and is not conducive to real-time object pose estimation.

In contrast, the method in this paper effectively solves the occlusion problem by feature fusion of different view. In addition, we do not need to ICP and other algorithms for point

TABLE I
DETECTION RESULTS IN OUR DATASET

metrics	method	bottom	handle1	handle2	handle3	handle4	top	mean
MAE(mm)	Direct	13.216	13	13.974	17.281	10.81	11.934	13.37
	Fine tuning	9.197	8.754	9.793	12.856	7.327	7.934	9.31
< 2cm(%)	Direct	92						
	Fine tuning	97.8						

TABLE II
DETECTION RESULTS OF DIFFERENT METHODS ON TOD DATASET

metrics	method	Ball	Bottle_1	Bottle_2	Cup	Mug_1	Mug_2	mean
MAE(mm)	DenseFusion	15.3	27.6	65.6	22.1	32.7	54.9	36.4
	Keypose	3.8	5.1	9.3	7.1	11.2	12.1	7.3
	Ours	4.64	5.16	6.24	6.13	6.78	7.32	6.04
< 2cm(%)	DenseFusion	78.8	18.1	9.1	54.4	8.1	0	28.08
	Keypose	100	99.7	91.4	95.3	87.2	87.1	93.5
	Ours	100	100	99.31	99.63	99.48	98.94	99.56
AUC	DenseFusion	84.7	72.3	47.5	77.8	67.3	48.1	66.3
	Keypose	96.1	94.9	90.7	92	88.6	87.8	91.7
	Ours	99.67	99.54	98.63	99.15	98.71	97.86	98.93

TABLE III
DETECTION RESULTS OF DIFFERENT METHODS ON YCB-VIDEO DATASET

metrics	method	Power_drill	Scissors	Large_clamp	Mug	mean
< 2 cm	PointFusion	22.8	35.9	50	99.8	52.1
	PoseCNN	99.6	95.6	74.9	99.8	92.5
	DenseFusion	98.7	100	79.2	100	94.5
	Ours	99.3	97.31	96.4	99.6	98.2
AUC	PointFusion	71.5	76.7	65.9	94.2	77.1
	PoseCNN	98.2	91.7	75.2	95	90
	DenseFusion	96	95.2	72.9	97.1	90.3
	Ours	97.3	96.5	95.9	96.2	96.5

cloud matching, but only the spatial information between the 3D key points to obtain more accurate object pose, so the algorithm has a faster detection speed to meet the needs of real-time detection.

D. The effect of the number of views on the detection results

In Fig. 8(a), we compare the effects of different fusion methods on the 3D keypoint detection for different numbers of camera views. “DLT” is the method mentioned in Section III. “sum aggregation” denotes the 3D feature map F_{3D}^{sum} obtained by directly summing the feature maps F_n from different camera views, which can be expressed as

$$F_{3D}^{\text{sum}}(i_x, i_y, i_z)_k = \sum_N (F_n\{P_{c_n}(i_x, i_y, i_z)\}_k). \quad (10)$$

In “softmax aggregation”, we first calculate the softmax of each view relative to all views to obtain the corresponding distribution coefficients,

$$\omega_n^{V'} = \frac{\exp F_n(\{P_{c_n}(i_x, i_y, i_z)\}_k)}{\sum_N \exp F_n(\{P_{c_n}(i_x, i_y, i_z)\}_k)}, \quad (11)$$

Then, the feature maps of each view are summed using the corresponding coefficients to obtain the fused 3D voxel feature maps, which can be formulated as

$$F_{3D}^{\text{vol}}(i_x, i_y, i_z)_k = \sum_N \omega_n^V \cdot (F_n\{P_{c_n}(i_x, i_y, i_z)\}_k). \quad (12)$$

“conf aggregation” is calculated using Eq. (3). It can be seen that all methods improve the accuracy of keypoint detection as the number of camera views increases, but when the number of the camera views are more than 4, the detection accuracy of the network model will no longer change significantly.

In Fig. 8(b), we tests the detection model on a Tesla V100 GPU and records the inference time for different submodules. When the number of camera views increases, its effect on the unprojection and 3D convolution is almost negligible. However, the inference time increases gradually for the 2D keypoint detection module of all views, so the time complexity of the algorithm is $O(N)$, N denotes the number of camera views. When the number of camera views greater than 6, the inference speed of the model is still within an acceptable range compared to the time for robot grasping process.

In summary, according to the results in Fig. 8, the method in this paper uses four camera views (corresponding to a real grasping scenario in which the camera at the end of the robot takes four pictures of different positions during robot motion)

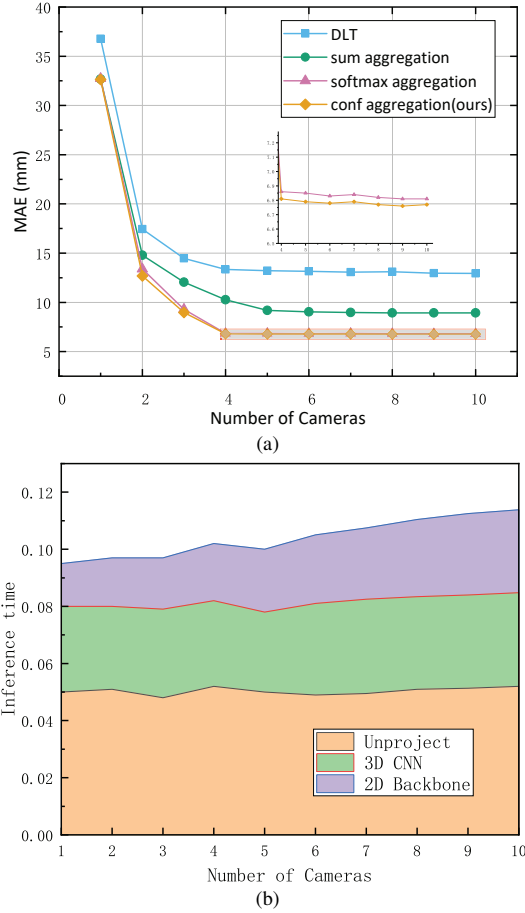


Fig. 8. The effect of the number of views on the detection results.

to achieve a high quality of object keypoint detection with guaranteed detection speed.

V. CONCLUSION

The object pose estimation of current methods has poor generalization performance and is not suitable for grasping scenes with complex environments. In this paper, we proposed a multi-view object keypoint detection algorithm, which takes color pictures with corresponding camera poses of different views as input, and outputs 3D keypoint pose information and the detection confidence of each view. Through a series of experiments comparing different datasets and methods, it can be seen that the high detection accuracy of our method, and it can also ensure the fast detection of the pose of the target object.

REFERENCES

- [1] J. Niu, C. Liu, Y. Gao, and M. Qiu. Energy efficient task assignment with guaranteed probability satisfying timing constraints for embedded systems. *IEEE Trans. on Para. and Dist. Systems*, 25(8):2043–2052, 2013.
- [2] J. Niu, Y. Gao, M. Qiu, and Z. Ming. Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability. *Journal of Para. and Dist. Computing*, 72(12):1565–1575, 2012.

- [3] W. Dai, L. Qiu, A. Wu, and M. Qiu. Cloud infrastructure resource allocation for big data applications. *IEEE Trans. on Big Data*, 4(3):313–324, 2016.
- [4] R. Lu, X. Jin, S. Zhang, M. Qiu, and X. Wu. A study on big knowledge and its engineering issues. *IEEE Trans. on Knowledge and Data Eng.*, 31(9):1630–1644, 2018.
- [5] Meikang Qiu, Keke Gai, and Zenggang Xiong. Privacy-preserving wireless communications using bipartite matching in social big data. *Future Generation Computer Systems*, 87:772–781, 2018.
- [6] M. Qiu and J. Li. Real-time embedded systems: optimization, synthesis, and networking. *CRC Press*, 2011.
- [7] Zhiguo Zhang, Jingqi Wu, Jing Deng, and Meikang Qiu. Jamming ack attack to wireless networks and a mitigation approach. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pages 1–5. IEEE, 2008.
- [8] M. Zhu, X.Y. Liu, F. Tang, M. Qiu, R. Shen, W. Shu, and M.Y. Wu. Public vehicles for future urban transportation. *IEEE Trans. on Intelligent Transportation Systems*, 17(12):3344–3353, 2016.
- [9] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362, 2018.
- [10] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6D pose object detector and refiner. In *IEEE/CVF Int'l Conf. on Computer Vision*, pages 1941–1950, 2019.
- [11] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.
- [12] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *IEEE/CVF Conf. on Comp. Vision and Pattern Recog.*, pages 4561–4570, 2019.
- [13] Lihao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Real-time 3D hand pose estimation with 3D convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):956–970, 2018.
- [14] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. KPAM: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019.
- [15] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 10059–10066, 2020.
- [16] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 3482–3489, 2018.
- [17] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [18] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *European Conf. on Computer Vision (ECCV)*, pages 385–400, 2018.
- [19] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-poseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *IEEE conf. on computer vision and pattern Recognition*, pages 5079–5088, 2018.
- [20] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *IEEE/CVF conf. on computer vision and pattern recognition*, pages 11602–11610, 2020.
- [21] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019.
- [22] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 244–253, 2018.