A Self-Supervised Learning-Based 6-DOF Grasp Planning Method for Manipulator

Gang Peng, *Member, IEEE*, Zhenyu Ren, Hao Wang, Xinde Li, *Senior Member, IEEE*, and Mohammad Omar Khyam

***Abstract*— To realize a robust robotic grasping system for unknown objects in an unstructured environment, large amounts of grasp data and 3D model data for the object are required; the sizes of these data directly affect the rate of successful grasps. To reduce the time cost of data acquisition and labeling and increase the rate of successful grasps, we developed a self-supervised learning mechanism to control grasp tasks performed by manipulators. First, a manipulator automatically collects the point cloud for the objects from multiple perspectives to increase the efficiency of data acquisition. The complete point cloud for the objects is obtained using the hand-eye vision of the manipulator and the truncated signed distance function algorithm. Then, the point cloud data for the objects are used to generate a series of six-degrees-of-freedom grasp poses, and the force-closure decision algorithm is used to add the grasp quality label to each grasp pose to realize the automatic labeling of grasp data. Finally, the point cloud in the gripper closing area corresponding to each grasp pose is obtained and used to train the grasp-quality classification model for the manipulator. The results of performing actual grasping experiments demonstrate that the proposed self-supervised learning method can increase the rate of successful grasps for the manipulator.**

***Note to Practitioners*—Most of the existing grasp planning methods of the manipulator are based on public datasets or simulation data to train model algorithms. Owing to the limited types of objects, the limited amount of data in the public datasets, and the lack of real sensor noise in the simulation data, the robustness of the trained algorithm model is insufficient, and it is difficult to apply to unstructured production environments. To solve the above problems, we propose a 6-DOF capture planning method based on self-supervised learning and introduce a self-supervised learning mechanism to solve the problem of grasp data acquisition in real scenes. The manipulator automatically collects object data from multiple perspectives, performs desktop-level 3D reconstruction, and finally uses the force-closure decision algorithm to automatically label the data in order to realize automatic acquisition and labeling of the grasp data in a real scenario. Preliminary experiments show that this method can obtain high-quality grasp data and can be applied to grasp operations in real multi-target and cluttered environments. However, it has not been tested in actual production environments. This paper focuses on the data acquisition module in the 6-DOF grasp planning framework. In future research, we will design a more efficient grasp planning module to improve the grasp efficiency of the manipulator.**

***Index Terms*—Manipulator, grasp planning, self-supervised learning, 3D reconstruction.**

I. Introduction

IN THE field of robotic arms, research in such areas as gripping, button operation and object propulsion [1] is popular. These studies often use visual servoing methods [2] for object manipulation. In particular, grasp planning based on visual information is essential to the development of intelligent robots. Nevertheless, there are still some challenges that must be overcome: 1) In a scene where multiple objects are stacked, the time to determine a feasible grasping posture is undesirably lengthy; 2) The geometric shapes of possible target objects are diverse and irregular, and a large amount of grasp data is required to train the algorithm; 3) If the robustness of the grasp planning algorithm is insufficient, changes in working environment will lead to a significant decline in the success rate of grasp.

Conventional algorithms for grasp models use a template matching algorithm and 3D model of the object to accurately calculate the pose of the object and then perform the grasp. Kehoe *et al.* [3] first constructed a 3D model of the target object and then used the Graspit! toolkit to build a grasp database for the model. In their method, a series of object models and corresponding feasible grasps are stored in the database. Then, the database is searched for a template that matches the point cloud for the target object; finally, the target object is grasped according to the preset grasp method. Owing to steady advancements in deep learning technology, Xiang *et al.* proposed an end-to-end object pose estimation network (PoseCNN) [4], which used three neural network branches to realize object pose estimation; it also enhanced the robustness and increased the accuracy of object position estimation algorithms, which could be used to guide the trajectory of grasp-purposed manipulators. In response to the

Manuscript received 24 August 2021; revised 12 October 2021; accepted 29 October 2021. Date of publication 2 December 2021; date of current version 13 October 2022. This article was recommended for publication by Associate Editor C. Yang and Editor K. Saitou upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 91748106 and in part by the Hubei Province Natural Science Foundation of China under Grant 2019CFB526. *(Zhenyu Ren is co-first author.) (Corresponding authors: Hao Wang; Xinde Li.)*

Gang Peng, Zhenyu Ren, and Hao Wang are with the Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: penggang@hust.edu.cn; renzhenyu@hust.edu.cn; wa_haoo@hust.edu.cn).

Xinde Li is with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: xindeli@seu.edu.cn).

Mohammad Omar Khyam is with the School of Engineering and Technology, Central Queensland University, Melbourne, VIC 3000, Australia (e-mail: m.khyam@cqu.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2021.3128639.

Digital Object Identifier 10.1109/TASE.2021.3128639

shortcomings of poor real-time performance of PoseCNN, Wang *et al.* proposed an iterative dense fusion network model (DenseFusion) [5], which makes full use of the two complementary data sources of color and depth; this greatly improves the speed of object poses estimation. Peng *et al.* proposed a pixel voting network (PVNet) [6], which directly uses RGB images for object pose estimation. This method uses the local information of the visible part of the object to extract the key points of the object of the image; it then uses the direction vector of each pixel of the visible part of the object of the key point as a feature and predicts the posture of the object of the neural network model. Through the extraction of key points and the calculation of direction vectors, this method can well estimate the pose of objects of complex situations such as occlusion and truncation, and the real-time performance is very good. The above-mentioned methods rely on 3D model data for the target objects and are still limited in their robustness and ability to enable real-time adjustments; moreover, they cannot be applied to grasp unknown objects.

Model-free grasp optimization algorithms focus on using the geometric information contained in the object's 3D point cloud to perform grasp planning and do not require accurate estimation of the object's pose. Regarding the robotic grasp process, this type of method does not require an accurate 3D model of the target object, which can be used to generate high-quality grasp poses for unknown objects. For example, [7] proposed a real-time grasping detection method, the main idea of the method is to perform grasping of novel objects in a typical RGB-D scene view. However, the method only performed a planar object grasping study, the grasping angle has some limitations and is not suitable for spatial 6-DOF grasping. Another example of model-free object grasp technology is the grasp pose detection (GPD) method proposed by Gualtieri *et al.* [8], which generates a series of candidate grasp poses by using the 3D point cloud of the object and geometric information on the parallel two-fingered gripper at the end of the manipulator and creates classification labels through the implementation of a force-closure analysis algorithm; the grasp pose quality is then classified using a convolutional neural network (CNN). With the development of cloud computing and big data technology, Mahler *et al.* proposed the Dex-Net [9], [10] series of datasets and related algorithms to enable robust grasp planning for the manipulator. Dex-Net researchers collected 10,000 independent 3D object models and used grasp wrench space analysis to create grasp pose classification labels. They then employed cloud computing technology to train a grasp-quality CNN, which ultimately yielded a robust grasp pose classification model. To realize the closed-loop capture of unknown objects by the robotic arm, Morrison *et al.* [11] proposed a pixel-level capture detection method. Its algorithm model is similar to the semantic segmentation network architecture, the input is a single-channel depth image, and the cornell grasp data set [12] is used for model training. First, a convolutional layer is used to extract the features, and then the transposed convolutional layer is used to achieve upsampling of the feature map. Finally, three mask images of the same size as the input image are obtained, which respectively represent the grasp ability, grasping angle, and grasping width of each pixel. This method can directly predict the best grasp point in the picture, and it can also directly return to the best grasp rectangle at the grasp point; hence, it runs extremely fast. To achieve robust capture of unknown objects, all the above methods use convolutional neural networks to classify the quality of the grasping posture. Therefore, large-scale objects and capture data are required to ensure the robustness of the training capture quality classification model.

In order to solve the problem of obtaining large-scale crawling data, Levine *et al.* [13] used more than 800,000 actual crawling attempts on a robotic arm to train a large-scale convolutional neural network model to achieve real-time visual servo crawling; however, the large number of actual crawls led to a significant increase in cost. To avoid the extra workload and reduce the capital required for data acquisition, Mousavian *et al.* proposed the 6-DoF GraspNet [14] method, which uses a particle-based simulation technology to perform grasp training in a simulation environment. It uses a variational autoencoder to achieve grasp sampling, and uses PointNet [15] to achieve grasp quality evaluation. In addition, a refinement module was designed through the cooperation of the refinement module and evaluation module, and iterative optimization based on the existing grasping posture was performed to quickly obtain a higher quality grasping posture. This method achieves the grasp of unknown objects to a certain extent; however, it only uses simulation data to train the model algorithm, so the robustness and generalization ability of the model algorithm needs to be improved.

From the above analysis of existing grasp planning methods for manipulators, it can be ascertained that, irrespective of whether it is a model-free algorithm, large-scale objects and grasp data are required to realize robust manipulator grasping performance. To solve the above problems, some data acquisition methods based on actual capture or simulation capture have appeared in recent years, but such methods still have some shortcomings. To further solve the problem of grabbing objects and obtaining data onto real scenes, avoid the actual grabbing operation of the robotic arm for the data acquisition process, reduce the cost of data acquisition, improve autonomous learning ability, and increase the rate of successful grasps for the manipulator, we developed a six-degrees-of-freedom (6-DOF) grasp planning method with self-supervised learning. Our contributions to the field can be summarized as follows:

1) The self-supervised learning mechanism allows the point cloud data for objects to be obtained, and it automatically labels the grasp data without requiring the manipulator to execute the actual grasp task; this reduces the time cost of data acquisition and labeling, and increases the level of automation in the grasping task;

2) The complete point cloud for target objects is obtained using the hand-eye vision of the manipulator and implementing a truncated signed distance function (TSDF) algorithm. A quality evaluation index for the object point cloud data is established, and an object point cloud dataset for real multi-target scenes is created;

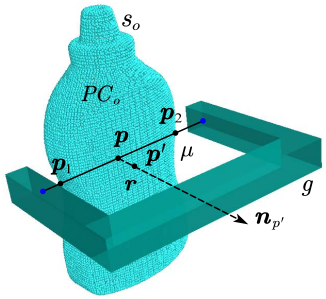


Fig. 1. Relationship between the grasp pose and complete point cloud for the object.

3) The implemented force-closure decision algorithm adds the grasp quality label to each 6-DOF grasp pose to enable automatic labeling of grasp data; this enables rapid creation of the grasp dataset.

II. Problem Statement

In our proposed method, the input of the grasp planning module is the single-view point cloud for the object; the feasible final 6-DOF grasp pose is obtained by means of grasp pose sampling and classification. If an unknown object O is given, the coefficient of friction $\mu \in \mathbb{R}$ between the object and gripper, the object's geometric and mass properties M_o, and the object's 6-DOF pose $\boldsymbol{W_o} \in \mathbb{R}^6$ are related to the grasp. Thus, $\boldsymbol{s_o} = (M_o, \mu, \boldsymbol{W_o})$ can be used to represent the state of the object. The task of the manipulator grasp planning module is to find a feasible final 6-DOF grasp pose $\boldsymbol{g} = (\boldsymbol{p}, \boldsymbol{r}) \in \mathbb{R}^6$, where $\boldsymbol{p} = (x, y, z) \in \mathbb{R}^3$ and $\boldsymbol{r} = \left(r_x, r_y, r_z\right) \in \mathbb{R}^3$ respectively specify the position and orientation of the grasp g.

It is assumed that the complete point cloud $PC \in \mathbb{R}^{3*N}$ containing N points for the object can be collected using a 3D scanner. Furthermore, to evaluate the quality of the final 6-DOF grasp pose $\boldsymbol{g}$, the metric $Q(\boldsymbol{s}, \boldsymbol{g_o}, \mu) \in \mathbb{R}$ is used to represent the grasp quality. Based on the above-described assumptions and definitions, the relationship between the grasp pose and the complete point cloud for the object can be described as is shown in Fig. 1. According to a point $\boldsymbol{p'}$ on the complete point cloud PC_o and the surface normal vector $\boldsymbol{n_{p'}}$ at its position, the position $\boldsymbol{p}$ and direction $\boldsymbol{r}$ of the grasp pose $\boldsymbol{g}$ can be determined. The object state s_o and the grasp pose $\boldsymbol{g}$ determine the positions of the contact points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, and then the quality $Q(\boldsymbol{s}, \boldsymbol{g_o}, \mu)$ can be determined according to the friction coefficient μ and the surface normal vector of the point cloud PC_o at the position of the contact point.

The state information $\boldsymbol{s_o}$ of the object O is hidden in the complete point cloud PC_o for the object O. After sampling, the grasp pose set $G = \{g_1, g_2 \cdots g_i\} \mid i \in \mathbb{N}$ can be obtained; then, the quality metric Q_i can be calculated using the coefficient of friction μ between the object and grasp pose $\boldsymbol{g_i}$. According to Q_i, the elements in the grasp pose set G can be sorted and subsequently used to guide the manipulator to execute grasp actions.

Depth cameras, which can directly obtain the 2.5D single-view point cloud data for an object, are typically employed as vision sensors in actual manipulator grasp tasks. Because it is difficult to obtain the complete state information $\boldsymbol{s_o}$ for the target object, it is impossible to calculate the grasp quality metric $Q(\boldsymbol{s}, \boldsymbol{g}, \mu)$. To solve this problem, it is necessary to learn a new quality metric $Q_\gamma(P, \boldsymbol{g}) \in \{c_0, c_1, \cdots\}$ through the process of training the model. This quality metric can only be calculated using the single-view point cloud P for the object, and the corresponding grasp pose $\boldsymbol{g}$; γ is a learning-based classification model of grasp quality, and $c_0, c_1, \cdots$ is a series of labels that characterize the quality of grasp pose $\boldsymbol{g}$. However, model training requires a large amount of 3D point cloud data for the object, as well as the corresponding 6-DOF grasp data and classification labels. Thus, to grasp unknown objects in an unstructured environment, there are three key obstacles to overcome: 1) How to obtain a large amount of point cloud data for objects in a real grasp scene; 2) How to generate classification labels for each grasp pose; and 3) How to design a grasp quality classification model.

To overcome the above-described obstacles, we constructed a self-supervised learning-based 6-DOF grasp planning algorithm framework for manipulators; it consists of two sub-modules, i.e., data acquisition and grasp planning sub-modules, as shown in Fig. 2. The grasp planning sub-module is responsible for generating a series of candidate grasp poses using the single-view point cloud, classifying and scoring the grasp poses using a grasp quality classification neural network model, and providing optimal grasp recommendations for the manipulator based on the grasp pose scores. The data acquisition sub-module obtains the complete point cloud for the target object by performing desktop-level 3D reconstruction; it then implements a force-closure decision algorithm to analyze the quality of the grasp pose to realize the automated acquisition and labeling of multi-target grasp data in a real grasp scene. This sub-module also provides the training data that allows the grasp planning sub-module to learn a new quality metric $Q_\gamma(P, \boldsymbol{g})$.

III. Grasp Data Acquisition

A. Desktop-Level 3D Reconstruction

In the proposed model, 3D reconstruction is an important part of the self-supervised learning-based data acquisition sub-module. Unlike large-scale 3D reconstruction algorithms such as BundleFusion [16], the 3D reconstruction algorithm proposed in this paper is purposed for desktop-level manipulator grasp scenes; additionally, the scene area to be reconstructed is smaller but higher accuracy of the reconstruction result is required.

In this model, the base coordinate system of the manipulator is the world coordinate system O_{world}; thus, the relationship between the end coordinate system O_{end} and base coordinate system O_{base} is ${}^b_e T = {}^w_e T$, which can be calculated by solving the forward kinematics equation for the manipulator. Then, the transformation matrix ${}^c_w T$ necessary to transform the world coordinate system to the camera coordinate system can be obtained as follows:

$$ {}^w_c T = {}^w_e T {}^e_c T \tag{1} $$

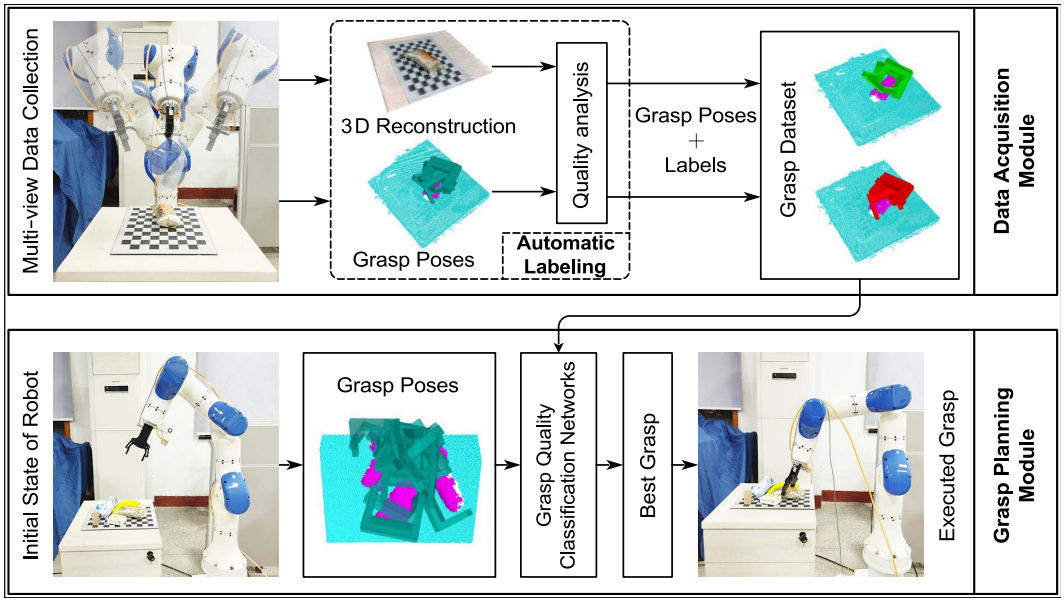


Fig. 2. Framework of proposed self-supervised learning-based 6-DOF grasp planning algorithm for manipulators.

In the above formula, ${}^{e}_{c}T$ represents the transformation between the camera coordinate system O_{camera} and end coordinate system O_{end}, i.e., the hand-eye transformation matrix, which can be obtained by applying hand-eye calibration to the manipulator. Using a camera pinhole imaging model, the relationship between a point $\boldsymbol{p}(u, v)$ in the depth image coordinate system and the corresponding point $\boldsymbol{P}(X_w, Y_w, Z_w)$ in the world coordinate system can be described as

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left({}^{w}_{e}T {}^{e}_{c}T \right)^{-1} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2}$$

In the above formula, f_x, f_y, u_0, and v_0 are the inherent properties of the camera, which can be obtained by calibrating the camera's internal parameters; Z_c is the depth value at point $\boldsymbol{p}$.

Our method entails first collecting the depth image set $\{img_1, i\ mg_2, \cdots img_N\}$ for the scene from N perspectives, and then calculating the camera pose $pose = {}^{c}_{w}T$ for each image frame to obtain the camera pose set $\{\ pose_1, pose_2, \cdots pose_N\}$. Then, Equation (2) is applied to convert the depth image data for each frame into a 3D point cloud PC_i; simultaneously, an improved TSDF algorithm [17] is used for point cloud data fusion to obtain the complete point cloud PC.

As a result of installing the camera at the end of the manipulator, and applying the forward kinematics and hand-eye transformation matrix for the manipulator to calculate the real-time pose of the camera, the time-expensive point cloud matching process that is necessary in conventional large-scale 3D reconstruction algorithms is no longer necessary; moreover, the camera pose estimation accuracy can be increased, thus increasing the dimensional accuracy of the reconstructed point cloud to allow the reconstructed complete point cloud to more accurately express the geometric information that describes objects in the scene.

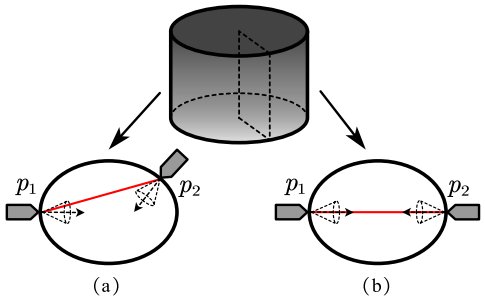


Fig. 3. Schematic illustration of grasp force closure.

B. Automated Grasp Pose Classification

In the process of grasp data acquisition, $Q_{fc} = Q(\boldsymbol{s}, \boldsymbol{g}, \mu)$ is applied as the grasp quality metric to realize automated grasp pose classification. This quality metric is calculated based on the mechanical relationship between the target object and gripper. The calculation of this mechanical relationship is called force-closure analysis. "Force closure" implies that the gripper, by means of making contact with the target object, can apply a force on the object to counteract other forces acting on the object, thereby ensuring that there will be no slippage at the contact point when the object is grasped.

Because object grasping with parallel two-finger grippers involves only two contact points, according to the Nguyen theorem proposed in [18], the necessary and sufficient condition for force closure is that the line connecting the contact points should be in the friction cone at the two contact points at the same time. In Fig. 3(a), the line connecting the two contact points is inside the friction cone at point $\boldsymbol{p_1}$ but not inside the friction cone at point $\boldsymbol{p_2}$; thus, the grasping action does not

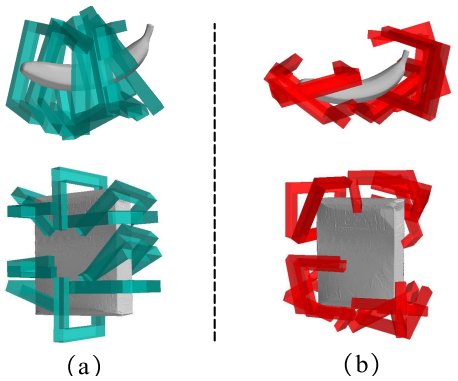


Fig. 4. Illustration of grasp pose classification.

satisfy the condition for force closure. In Fig. 3(b), the line connecting the two contact points passes through the centers of both friction cones; thus, this grasping action satisfies the condition for force closure.

Given two contact points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ in a 3D space, the surface normal vectors $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ at the two contact points and the static friction coefficient μ can be obtained. Then, the unit vector from point $\boldsymbol{p_1}$ to point $\boldsymbol{p_2}$ can be expressed as

$$\hat{v} = \frac{\boldsymbol{p_2} - \boldsymbol{p_1}}{\|\boldsymbol{p_2} - \boldsymbol{p_1}\|_2} \mid \boldsymbol{p_1}, \quad \boldsymbol{p_2} \in \mathbb{R}^3 \tag{3}$$

The angle between $\boldsymbol{n_1}$ and $\hat{v}$ is $\alpha_1 = \cos^{-1}(\boldsymbol{n_1} * (-\hat{v}))$, and the angle between $\boldsymbol{n_2}$ and $\hat{v}$ is $\alpha_2 = \cos^{-1}(\boldsymbol{n_2} * (\hat{v}))$. The half-vertex angle of the friction cone at the contact point is $\beta = \tan^{-1}(\mu)$; according to the Coulomb friction model, when $\alpha_1 < \beta$ and $\alpha_2 < \beta$, the force-closure condition is satisfied, and $Q_{fc} = 1$; otherwise, $Q_{fc} = 0$.

We used the method proposed in [8] to generate the grasp poses and applied the scoring mechanism proposed in [12] to improve the classification scheme for the grasp pose. For a certain grasp pose $\boldsymbol{g}$, and beginning at a value of 3.0, the coefficient of friction μ is gradually reduced until $\boldsymbol{g}$ does not satisfy the force-closure condition; the smallest coefficient of friction μ that satisfies the force-closure condition is recorded as the score for $\boldsymbol{g}$. Fig. 4(a) shows the grasp poses that satisfy the force-closure condition when $\mu = 0.4$, and Fig. 4(b) shows the grasp poses that satisfy the force-closure condition when $\mu = 2.0$. A smaller coefficient of friction corresponds to a higher quality of grasp poses that satisfy the force-closure condition.

C. Self-Supervised Learning Mechanism

The grasp dataset should not only contain object point cloud data but also a series of grasp poses and corresponding labels. However, it is inefficient to employ the manipulator in actual grasping tasks for the purpose of adding classification labels for grasp poses. Thus, to realize automated annotation of manipulator grasp poses, it is best to apply the force-closure condition to analyze the reliability of the grasp pose; however, the force-closure condition can only be established when the complete geometry of the object and the positions of the contact points are known; this means that force-closure analysis for the grasp pose can only be performed when the point cloud for the object is complete.

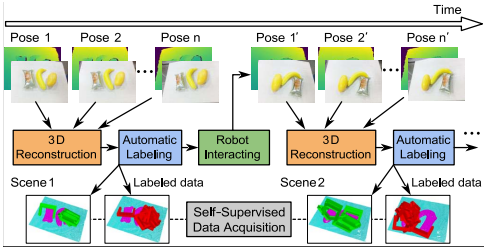


Fig. 5. Overview of the proposed self-supervised data acquisition sub-module.

At present, the majority of the most frequently employed 6-DOF motion planning methods for manipulator grasping require object datasets such as BigBIRD [19] and YCB [20], which include a series of single-view and complete point clouds for objects; additionally, 3D scanners and professional point cloud acquisition systems are employed in the production process. Nevertheless, these object datasets have the following problems: 1) The amount of data is limited; 2) The types of objects are limited; 3) The objects are not in a real grasp scene; and 4) Only the point cloud data for individual objects are included, even though there are typically multiple objects in the grasp scene.

To overcome the challenges related to collecting grasp data in real scenes, inspired by the work of [21], [22] and others, we designed a self-supervised learning-based data acquisition sub-module; it is shown in Fig. 5. First, the initial state of the objects on the desktop is manually set; then, the manipulator automatically collects image data, performs desktop-level 3D reconstruction, and autonomously pushes the objects on the desktop according to the reconstructed scene information; data collection and 3D reconstruction of the next scene is continued until a preset amount of grasp data is collected.

Our proposed self-supervised learning-based data acquisition method entails implementation of a desktop-level 3D reconstruction algorithm to obtain the complete point cloud for objects, followed by the realization of the automatic annotation of grasp poses through the implementation of a force-closure analysis algorithm. This method has the following advantages: 1) The 6-DOF grasp planning method for the manipulator is not dependent on existing object datasets; 2) Grasp data can be collected from multi-object and stacked object scenes; 3) Data can be automatically collected and labeled in a real capture scene, thereby allowing the manipulator to learn autonomously.

IV. Deep Learning-Based Grasp Quality Classification

The quality of the grasp pose is determined by the mechanical relationship between it and the target object. We regard the point cloud information in the closing area of the gripper as a representation of the mechanical relationship between the grasp pose and the object; it is employed as the input of the grasp quality model γ . Following input, the classification problem of grasp pose quality is transformed into a classification

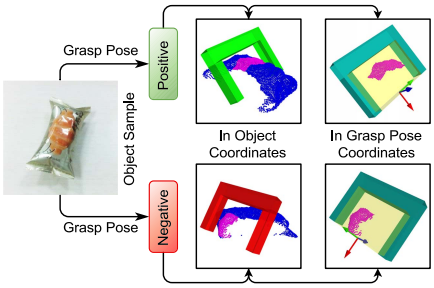


Fig. 6. Method for obtaining the point cloud for the gripper closing area.

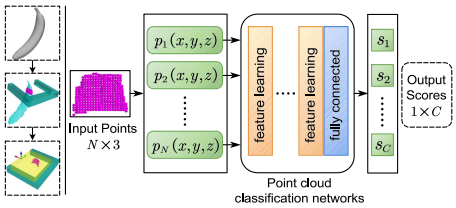


Fig. 7. Grasp quality classification scheme based on deep learning-based point cloud analysis.

problem of the point cloud for the closed area of the gripper. The method of obtaining the point cloud for the closing area of the gripper is illustrated in Fig. 6, where green indicates a feasible grasp, red indicates an infeasible grasp, light yellow indicates the closing area, and purple indicates the point cloud for the closing area.

Several deep learning models for point cloud analysis that are based on PointNet [15] take the original point cloud as the input; most of them have excellent feature-extraction capability and can learn the point cloud feature information through training. This typically affords superior robustness and a very high forward propagation speed; thus, this approach is advantageous for tasks that require classification of sparse point clouds or missing point clouds. The spatial domain of the point cloud in the gripper closing area is small. Thus, applying deep learning to classify the point cloud for the gripper closing area can solve problems related to force-closure analysis not being applicable for the classification of single-view point clouds; moreover, this approach can ensure the speed and accuracy of point cloud classification, thus ensuring a high speed and success rate of the manipulator grasp planning task.

The proposed method of deep learning-based point cloud analysis for grasp quality classification is shown in Fig. 7. First, the point cloud for the closing area of the gripper in the grasp pose coordinate system is obtained; then, the point cloud is taken as the input for the point cloud classification networks; finally, the scores for each category within this point cloud are output. The classification score of the point cloud for the gripper closing area can be used to determine whether the grasp pose associated with this point cloud is reliable, and the grasp pose can be sorted according to this score. Finally, the grasp pose with the highest score is selected for initiation by the manipulator for the actual grasping task.

V. Experimental Results and Analysis

To verify the effectiveness and advantages of the proposed method, a real-scene data acquisition experiment was carried out; then, a simulated grasp pose classification experiment was carried out; finally, a real-scene grasp experiment was carried out. The computer used for these experiments was configured as follows: Intel Core i5-8300H CPU, 2.3 GHz, GTX1060 GPU, and 16 GB of RAM.

Fig. 8. Objects used for dataset production and actual grasping.

TABLE I
Success Rates of Single-Target Grasp Experiments

Dataset	Model	Fruit	Box	Column	Avg
YCB	GPD (12ch)	63.3%	68.3%	73.3%	68.3%
	PointNet	81.6%	86.7%	91.7%	86.6%
	PointNet++	75.0%	81.7%	90.0%	82.2%
SSG	GPD (12ch)	61.7%	70.0%	78.3%	70.0%
	PointNet	85.0%	90.0%	93.3%	**89.4%**
	PointNet++	78.3%	83.3%	91.6%	84.4%

We used a self-developed 7-DOF non-biased S-R-S manipulator for the real-scene data collection and grasp experiments. The end of the robot was equipped with a parallel two-finger gripper and an Intel Realsense D435i depth camera. The maximum distance between the two fingers of the gripper was approximately 7 cm.

We selected 18 common objects for the experiments; they were divided into the following three categories: fruits, boxes, and columns. The left photograph in Fig. 8 shows the objects used for dataset production, and the right photograph shows the unknown objects used for grasping. Because the operational range of the gripper was limited, the sizes of the selected objects also had to be small; this condition ensured that each object had a clampable part with a width that ranged between 1 and 6 cm.

A. Data Acquisition Experiment

Our grasp planning method analyzes the quality of the grasp pose by utilizing the geometric information on the object; thus, it is very important to ensure consistency between the geometric features of the single-view point cloud and the complete point cloud for the object dataset, which is guaranteed by the accuracy of the 3D reconstruction obtained via the data acquisition process.

We conducted a data acquisition experiment to 1) assess the degree of similarity between the single-view point cloud for the object collected via this method, and the complete point cloud; and 2) evaluate the quality of the obtained object data. In the experiment, a total of 450 single-view point clouds and corresponding complete point clouds were collected for the nine objects shown in the left photograph of Fig. 8; various combinations of these objects were applied, and 900 single-view point clouds and corresponding complete

TABLE II
DETAILED EXPERIMENTAL DATA FOR A SET OF SINGLE-TARGET GRASP TASKS

Categories	Objects	PointNet					PointNet++					GPD (12 ch)				
		Trials				SR	Trials				SR	Trials				SR
		#1	#2	#3	#4		#1	#2	#3	#4		#1	#2	#3	#4	
Fruit	Carambola	√	√	×	√	3/4	×	√	√	×	2/4	×	√	×	√	2/4
	Kiwi	√	√	√	√	4/4	√	×	√	√	3/4	√	√	√	√	4/4
	Pear	×	√	√	√	3/4	√	√	√	√	4/4	×	√	×	×	1/4
	Overall					83.3%					75.0%					58.3%
Box	Pill box	√	√	√	√	4/4	√	√	√	×	3/4	×	√	×	√	2/4
	Paper napkin	√	√	√	√	4/4	√	√	√	√	4/4	×	√	√	√	3/4
	Wafer	√	×	√	√	3/4	√	√	×	√	3/4	√	√	×	√	3/4
	Overall					91.6%					83.3%					66.7%
Column	Cylinder	√	√	√	√	4/4	√	√	√	√	4/4	√	√	√	×	3/4
	USB cable	×	√	√	√	3/4	×	√	√	√	3/4	√	√	×	×	2/4
	Cylindrical block	√	√	√	√	4/4	√	√	√	√	4/4	√	√	√	√	4/4
	Overall					91.6%					91.6%					75.0%
Overall						88.9%					83.3%					66.7%

Each model was trained on the SSG dataset;SR indicates the rate of successful grasps.
√:the trial was successful; ×:the trial was not successful; ×:none of the grasps were kinematically feasible.

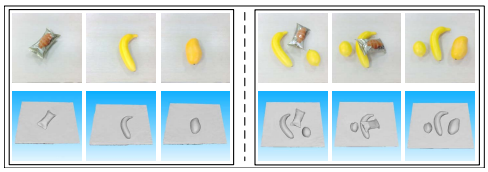


Fig. 9. Desktop-level 3D reconstruction results.

point clouds for these combinations were collected and added to a self-supervised grasp (SSG) dataset.

To ensure the accuracy and real-time performance of the camera pose calculations, an ROS bag tool was used to save depth images and coordinate system information as the manipulator moved in real-time. After the manipulator motion was completed, the bag file was parsed to obtain the coordinate system information for the camera pose calculations, and the corresponding depth images were obtained by referencing the timestamps; unmatched depth images were discarded. Finally, a series of acquired depth images and corresponding camera pose data were used to perform desktop-level 3D reconstruction. Fig. 9 shows the experimental results of implementing the proposed desktop-level 3D reconstruction method.

Ideally, the single-view point cloud for an object should be a subset of its complete point cloud. To quantitatively evaluate the quality of the object data in our SSG dataset, an object data precision index acc_{Match} was defined to quantify the degree of similarity between the single-view point cloud and complete point cloud for the objects in the dataset.

$$acc_{Match} = \frac{point\_num_{Match}}{point\_num_{All}} \tag{4}$$

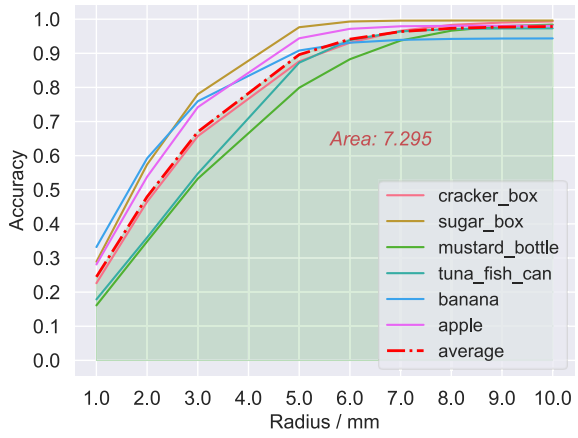
We have defined the accuracy of object data as the ratio of the number of matched points between a single-view point cloud and the corresponding complete point cloud, to the total number of points in the single-view point cloud, where the single-view point cloud and complete point cloud are represented in the same reference coordinate system. Ideally, all points in the single-view point cloud should be able to find matching points in the complete point cloud; thus, a larger value indicates higher reconstruction accuracy. We found the matching points by creating and implementing a KD-tree index for the complete point cloud, and then by traversing every point in the single-view point cloud. If at least one point in the complete point cloud could be found within a sphere with a radius r at a certain point, this point was determined to be a matched point.

Regarding the YCB dataset, the complete point cloud was obtained by using a 3D scanner, whereas the single-view point cloud was obtained by using multiple depth cameras. To unify the reference coordinate systems for the single-view point cloud and complete point cloud, an iterative closest point (ICP) algorithm was implemented to register the single-view point cloud to the complete point cloud coordinate system. Six independent objects and 50 single-view point clouds from the YCB and SSG datasets were respectively selected to enable object data accuracy comparison; the average reconstruction accuracies were calculated by applying different search radii.

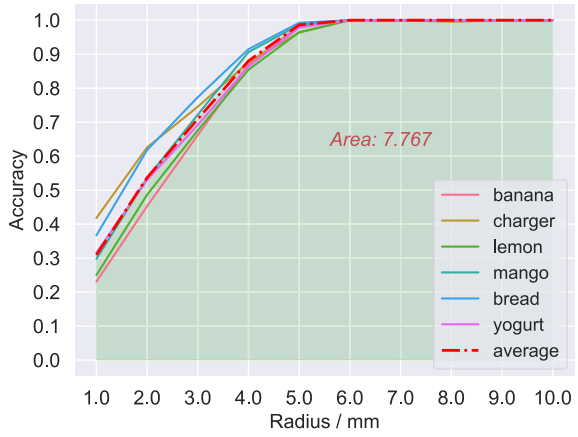
The object data accuracy results are shown in Fig. 10. The horizontal axis in the figure represents search radius (r) values; the red dashed line represents the average accuracy results for all objects, and the remaining curves show the average accuracy results for individual objects. The light green area between the red dashed line and the horizontal axis reflects the overall reconstruction accuracy. The overall accuracy of the object data generated from our SSG dataset was 6.47% higher than that generated using the YCB dataset; this proves that our method can be used to obtain higher quality object data. This is because the single-view point cloud for the YCB dataset was collected using multiple depth cameras, and the high accuracy of the internal parameter calibration for each camera, as well as the coordinate transformation matrix required to synchronize camera data is difficult to guarantee. However, our method avoids such a complicated sensor calibration problem.

B. Simulated Experiment

To further verify the effectiveness of the proposed grasp data acquisition method, 1,500 grasp poses and their corresponding labels were generated for the aforementioned nine objects; nine objects from the YCB dataset were also selected for comparative analysis, and the same number of grasping poses



(a) YCB dataset



(b) SSG dataset

Fig. 10. Object data accuracy results.

and labels were generated for them. The experiment was carried out as follows: 1) Create a coefficient of friction list, i.e., $list_\mu = \{3.0, 2.0, 1.7, 1.4, 1.3, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$; 2) Generate 100 grasp poses for each coefficient of friction in $list_\mu$; 3) To increase the discriminability between positive and negative samples, set thresholds $th_{good} = 0.45$ and $th_{bad} = 0.75$. Grasp poses with a μ less than or equal to $th_{good} = 0.45$ should be regarded as high-quality grasps, and grasp poses with a μ greater than or equal to $th_{good} = 0.45$ should be regarded as low-quality grasps. The labels were assigned according to the following formula:

$$label_g = \begin{cases} 1 & \mu_g \leq th_{\text{good}} \\ 0 & \mu_g \geq th_{\text{bad}} \end{cases} \tag{5}$$

To ensure equal numbers of positive and negative samples, 200 high-quality grasp poses for the nine objects and 200 randomly selected low-quality grasp poses were added to the dataset, with 20% being applied as the test set. Next, the geometric model of the gripper was used as a reference to process 50 single-view point clouds for the objects corresponding to each grasp pose, and the point cloud for the gripper closing area was extracted; the data were up-/down-sampled to 1,024 points. Thus, 180,000 sets of closing area point clouds and corresponding labels were generated for the

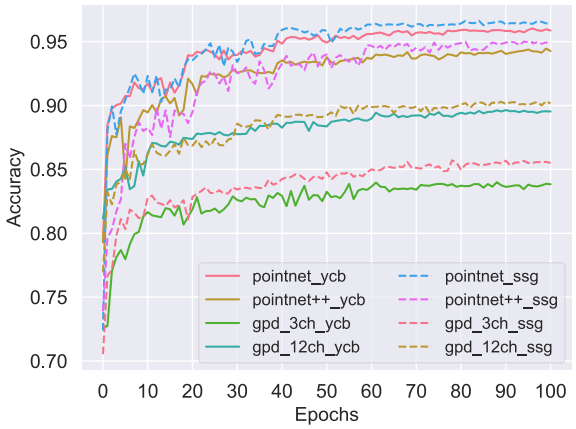


Fig. 11. Accuracy of each tested classification model. The best results are obtained using the SSG dataset and the network model is pointnet.

two datasets, and subsequently used to train the grasp quality classification networks.

Three network models, GPD [8], PointNet [15], and PointNet++ [23], were implemented in the simulation experiment to classify the point cloud in the gripper closing area. Between them, PointNet and PointNet++ are deep learning-based point cloud models, whereas GPD consists of conventional CNNs. The results of training each model on the YCB and SSG datasets are shown in Fig. 11.

Analysis of the experimental results (Fig. 11) revealed that the models trained on the SSG dataset were able to effectively learn, thereby proving the effectiveness of the proposed self-supervised learning-based method for manipulators. Additionally, the accuracy of each model trained on the SSG dataset tended to be higher than that of its counterpart trained on the YCB dataset. However, it should be noted that the test data were different; thus, it is not enough to prove that the proposed method can increase the accuracy of the classification model, and experiments must be carried out in real grasping scenes. It is also noteworthy that the classification accuracy achieved by applying a conventional GPD method to the two datasets was considerably lower than that obtained via the deep learning-based point cloud method; this confirms the feasibility of the proposed grasp quality classification method. Interestingly, the application of the proposed point cloud method to PointNet yielded the best classification results for the gripper closing area, even better than the improved version, PointNet++. This is because PointNet++ improved the ability of the network to extract localized point cloud information; this led to the network model paying too much attention to the localized point cloud information. This was a problem because the spatial domain of the point cloud for the gripper closing area was small; this means that there was not much localized information. Thus, the overall characteristics of the point cloud are more useful for classification.

C. Grasp Experiment

Our grasping pipeline is shown in Fig. 12. To ensure that the grasp task is performed with high efficiency and high success rate, we applied several preprocessing procedures to the data from the depth sensor. First, the depth range was

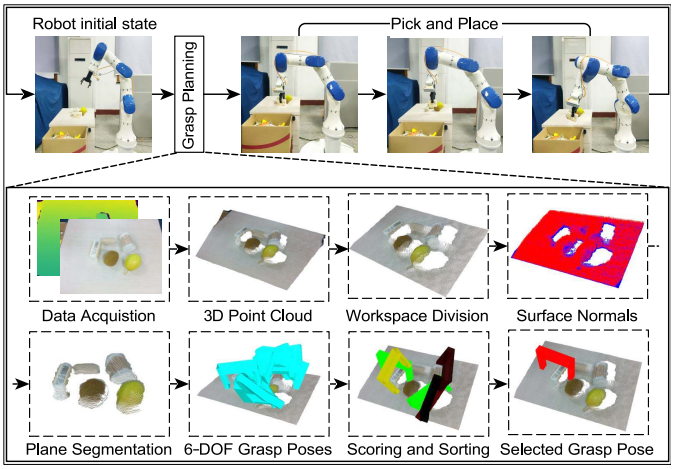


Fig. 12. Grasping pipeline. The most important of these are 6-DOF Grasp Poses and Scoring and Sorting. 6-DOF Grasp Poses are responsible for generating candidate crawls, and Scoring and Sorting is responsible for classifying the quality of candidate grasp.

limited to 0–0.6. Then, instead of applying the point cloud format in ROS, the depth images and internal parameters of the camera were used to generate organized point clouds to increase the speed of calculation of the surface normal of the point cloud. Next, the workspace was setup to only take points within the following ranges as input: $-0.2m < x < 0.2\ m$ and $-0.2m < y < 0.5\ m$. Finally, the RANSACE algorithm was applied to segment the desktop; the grasp points were uniformly sampled in the point cloud above the desktop.

1) Single-Target Grasp Experiment: In the single-target grasp experiment, only single-object data from the YCB and SSG datasets were used to train the grasp quality classification models, and five sets of grasp experiments were carried out for each trained model. Each set of experiments involved 36 grasps and nine objects from three different categories. A grasp was deemed to be successful if the 6-DOF grasp pose data output by the grasping planning algorithm resulted in the manipulator being able to grasp, lift, and hold the target object for 2 s while not touching other objects. The experimental results are summarized in Table I; detailed results for one set of experiments are provided in Table II. The models trained on our SSG dataset tended to outperform those trained on the YCB dataset; these results prove that the proposed method can increase the rate of successful grasps for a manipulator employed in a real scene. It should also be noted that, in the grasp experiments using fruit and other geometrically complex objects, the proposed PointNet-based method has a greater improvement than the conventional 6-DOF grasp planning method.

2) Multi-Target Grasp Experiment: To further verify the advantages of the proposed self-supervised learning-based method for the manipulator, SSG data for various combinations of objects were added to the training set, and the above-mentioned three network models were retrained for grasp experiments in multi-target and cluttered scenes. Six objects were selected as the grasp targets, and 20 sets of experiments were performed using the models trained on the YCB dataset and our SSG dataset. The goal of each set of experiments was to perform a grasp task 10 times with the time being recorded. The experimental results are summarized in Table III. The rate of successful (SR) grasps in the table refers to the ratio of the number of successful grasps to the total number of grasps after the desktop has been emptied or 10 attempts have been made. The completion rate (CR) refers to the ratio of the number of removed objects to the total number of objects after 10 grasp attempts, and the grasp preparation time (PT) refers to the time spanning the beginning of image acquisition to the determination of the optimal grasp.

TABLE III
MULTI-TARGET AND CLUTTERED SCENE GRASP EXPERIMENT RESULTS

Model	SR		CR		PT
	YCB	SSG	YCB	SSG	
GPD (12ch)	64.8%	66.6%	79.2%	82.5%	1.22 s
PointNet	82.6%	**86.2%**	87.5%	**92.5%**	1.37 s
PointNet++	79.3%	82.9%	85.0%	90.8%	3.43 s

The experimental results show that, in terms of the rates of success and completion, the proposed 6-DOF grasp planning algorithm is best utilized by the PointNet or PointNet++ network model for multi-target and cluttered scene grasp tasks; furthermore, comparison to the conventional 6-DOF grasp planning method confirmed significant improvements. However, it is important to note that the implementation of the proposed algorithm in the PointNet++ network model significantly reduces grasp planning efficiency, making it unsuitable for grasp tasks with stringent real-time constraints. In addition, the models trained on our SSG dataset outperformed those trained on the conventional YCB dataset; this proves that the proposed self-supervised learning-based method can achieve a higher rate of successful grasps for manipulators applied in multi-target and cluttered scenes.

VI. CONCLUSION

Objects in an unstructured environment have various shapes and sizes. To ensure the robustness of the 6-DOF grasp planning algorithm for the manipulator, we developed a self-supervised learning mechanism. In this study, it was demonstrated to autonomously guide the manipulator to collect and label data when applied for use in real grasping scenes, eliminating the need for actual grasping operations. We also defined an object data accuracy index to quantitatively evaluate the quality of object datasets. The experimental results revealed that the proposed method can be employed to obtain high-quality grasp data, increase the rate of successful grasps for manipulators in real scenes, reduce the computational cost of grasp data acquisition, and realize a self-learning-based grasp planning method for manipulators.

In future work, we plan to optimize the grasp planning module to further increase the grasp efficiency and success rate. Since our method only uses local point clouds for grasping quality classification, the accuracy may not be ideal, so we plan to add global point cloud information as part of the grasp quality classification networks' input to improve the robustness of the grasp planning algorithm.

REFERENCES

[1] C. Zeng, C. Yang, H. Cheng, Y. Li, and S.-L. Dai, "Simultaneously encoding movement and sEMG-based stiffness for robotic skill learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1244–1252, Feb. 2021.

[2] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of a robotic arm with visual fusion technology," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3822–3830, Sep. 2018.

[3] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the Google object recognition engine," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4263–4270.

[4] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot., Sci. Syst. Conf.*, Pittsburgh, PA, USA, Jun. 2018. [Online]. Available: http://www.roboticsproceedings.org/rss14/p19.html, doi: 10.15607/rss.2018.xiv.019.

[5] C. Wang *et al.*, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3343–3352.

[6] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.

[7] J. Zhang, M. Li, Y. Feng, and C. Yang, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools Appl.*, vol. 79, nos. 3–4, pp. 2427–2446, Jan. 2020.

[8] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 598–605.

[9] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5620–5627.

[10] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau4984. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aau4984, doi: 10.1126/scirobotics.aau4984.

[11] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robot., Sci. Syst. Conf.*, Pittsburgh, PA, USA, Jun. 2018. [Online]. Available: http://www.roboticsproceedings.org/rss14/p21.html, doi: 10.15607/rss.2018.xiv.021.

[12] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 705–724, 2015.

[13] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2017.

[14] A. Mousavian, C. Eppner, and D. Fox, "6-DOF GraspNet: Variational grasp generation for object manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2901–2910.

[15] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[16] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, p. 76, 2017.

[17] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1996, pp. 303–312.

[18] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.

[19] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "BigBIRD: A large-scale 3D database of object instances," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 509–516.

[20] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollár, "Benchmarking in manipulation research: Using the yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[21] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6D object pose estimation for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3665–3671.

[22] R. Jeong *et al.*, "Self-supervised Sim-to-real adaptation for visual robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2718–2724.

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. 30*, I. Guyon *et al.*, Eds., Long Beach, CA, USA, Dec. 2017, pp. 5099–5108. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html and https://dblp.org/rec/conf/nips/QiYSG17.bib

Gang Peng (Member, IEEE) was born in 1973. He received the Ph.D. degree from the Department of Control Science and Engineering, Huazhong University of Science and Technology (HUST), in 2002. Currently, he is an Associate Professor with the Department of Automatic Control, School of Artificial Intelligence and Automation, HUST. His research interests include intelligent robots, machine vision, multi-sensor fusion, machine learning, and artificial intelligence. He is also a Senior Member of the China Embedded System Industry Alliance, the China Software Industry Embedded System Association, and the Chinese Electronics Association; and a member of the Intelligent Robot Professional Committee of Chinese Association for Artificial Intelligence.

Zhenyu Ren was born in 1996. He received the bachelor's degree from the School of Automation, Hainan University, China, in 2018, and the master's degree from the Department of Automatic Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include intelligent robots and perception algorithms.

Hao Wang was born in 1996. He received the bachelor's degree from the School of Automation, Wuhan University of Technology, Wuhan, China, in 2019. He is currently a Graduate Student at the Department of Automatic Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan. His research interests are intelligent robots and perception algorithms.

Xinde Li (Senior Member, IEEE) was born in 1975. He received the Ph.D. degree in control theory and control engineering from the Department of Control Science and Engineering, Huazhong University of Science and Technology (HUST), in 2007. Thereafter, he joined the School of Automation, Southeast University, Nanjing, China, where he is currently a Professor and the Ph.D. Supervisor. From January 2012 to January 2013, he was a National Public Visiting Scholar at Georgia Tech Visit and Exchange for one year. From January 2016 to August 2016, he worked as a Research Fellow with the ECE Department, National University of Singapore. His main research interests include intelligent robots, machine vision perception, machine learning, human–computer interaction, intelligent information fusion, and artificial intelligence.

Mohammad Omar Khyam was born in 1988. He received the Ph.D. degree in electrical and electronics engineering from the University of New South Wales (UNSW) in 2015. Thereafter, he respectively engaged postdoctoral research with the National University of Singapore, Nanyang Technological University, Virginia Polytechnic Institute, and State University. Since July 2019, he has been with the School of Engineering and Technology, Central Queensland University, Australia. His main research interests include intelligent robots, machine vision perception, and artificial intelligence.