



A 3D pickup estimation method for industrial parallel robots based on point cloud simplification and registration

Yuting Zhang^{1,2} · Zongyan Wang^{1,2} · Xi Wang^{1,2} · Pei Gao^{1,2} · Menglong Li^{1,2}

Received: 4 August 2023 / Accepted: 20 June 2024 / Published online: 2 July 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

In industrial environments, cameras are heavily influenced by light and perspectives, making traditional vision-based parallel robots struggle with object-sorting tasks. Two-dimensional vision lacks depth information, which limits parallel robots to pick up objects at various heights. This paper proposes a 3D pickup estimation method for parallel robots based on point cloud simplification and registration for multi-objective pickup tasks. Firstly, a point cloud segmentation method based on the spatial domain is proposed to separate incomplete point clouds of object from the robot's workspace. The object dataset is generated by scanning complete point clouds of object using a 3D scanner. Secondly, according to the fast point feature histogram (FPFH) and the weight locally optimal projection (WLOP) algorithms, fusing the FPFH and WLOP (FF-WLOP) method is proposed to simplify the incomplete point cloud and extract more distinctive edge features of objects. The complete point cloud from the dataset is aligned with the simplified incomplete point cloud, and the calculated barycenter's coordinate information is given to the incomplete point cloud. Next, a dynamic weight singular value decomposition (D-SVD) hand-eye calibration method and an optimal projection point strategy are proposed to transform the barycenter coordinates of the object to the optimal pickup coordinates. Experimental results show that the point cloud registration error is 0.38 mm, the pickup rate is 92%, and the robot positioning error is 4.67 mm, which meets the basic pickup requirements.

Keywords Parallel robots · 3D point cloud · Hand-eye calibration · Robot pickup estimation · Optimal projection point

1 Introduction

With the continuous advancement of industrial robot automation production lines, machine vision and artificial intelligence technology have become widely used in industrial robots, particularly in the areas of robot identification, classification, and object pickup [1–3]. Parallel robots play a crucial role in food packaging, workpiece sorting, and intelligent assembly due to their good structural stability and fast pickup characteristics [4, 5]. Three-dimensional pickup estimation has emerged as an important topic in robotics, allowing for adaptive pickup of various objects through posture evaluation and grasp point analysis, thereby improving the success rate of pickups [6–10].

The 3D point cloud can express the three-dimensional characteristics of objects with high accuracy and environmental adaptability. Currently, 3D point cloud technology is most commonly utilized in mobile robots, particularly for vision-based SLAM mapping and obstacle avoidance in various industrial scenarios [11–13]. However, 3D point cloud pickup methods are still in the research stage and face challenges such as higher hardware requirements and restrictions in complex environments compared to 2D machine vision methods [14], while many studies have focused on point cloud pickup methods for collaborative robots in ideal laboratory environments. For example, Liu et al. [15] carried out robotic welding work on pipes through a real point cloud with the NSGA-II algorithm, which accurately located the position of the pipe to be welded by preserving the local features of the point cloud and fitting calculations. Zhuang et al. [16] proposed a point cloud segmentation method to estimate the 6D grasp posture information of objects, which combined depth images and point cloud information. Faria et al. [17] studied robot imitation of human behavior to achieve dexterous grasping of irregular objects and estimate

✉ Zongyan Wang
wangzongyan321@sina.com

¹ School of Mechanical Engineering, North University of China, Taiyuan 030051, China

² Shanxi Provincial Key Laboratory of Digital Design and Manufacturing, Taiyuan 030051, China

grasping position through object point cloud segmentation and modeling. These methods utilize advanced point cloud processing techniques, which provide a good enhancement of the pickup results, but the environment is more ideal and does not consider practical applications in industrial environments.

With the development of deep learning, point cloud segmentation algorithms have been newly improved, and the point cloud object segmentation algorithms of PointNet [18] and PointNet++ [19] have laid a new foundation for robot posture estimation. Liang et al. [20] utilized the PointNet algorithm for grasp posture estimation to form the robot gripper's posture directly on the 3D point cloud, which is computationally less intensive than traditional CNN methods. Ni et al. [21] proposed an end-to-end bit position estimation network based on PointNet++ using sparse point cloud and Ferrari-Canny multi-target pickup algorithm, the result of robot grasping accuracy is higher than PointNet. However, deep learning methods require dataset labeling and entail significant computational resources during training, making them less suitable for industrial production scenarios where robustness and stability are crucial [22–25].

Many industrial parallel robots rely on air-pumped suction cups as their end-effectors, specifically for object pickup on the upper surface or in the same direction. Zhang et al. [26] used a knowledge-learning suction region prediction method to analyze the suction force magnitude versus object barycenter for robotic pickups where the end-effector is a suction cup. Han et al. [27] used an unsupervised learning suction detection for suction cup pickup of aggregated objects using two convolutional neural networks for region of interest (ROI) extraction and suction point detection, respectively. The above methods are analyzed for suction cup pickup, and the positioning accuracy needs to be improved. Another key is that the quality of the point cloud directly affects the object positioning effect. Li et al. [28] proposed DPGNet, a point cloud completion network that utilizes a multi-resolution context encoder to gradually fuse the global information to achieve point cloud reconstruction. Zhou et al. [29] proposed a point cloud denoising method using nonlocal self-similar features to remove outliers and interfering points in the environment and maintain the outer contour features of the object through the height matrix. Point cloud reconstruction can restore localized data; it often differs significantly from the original point cloud in robot pickup and positioning tasks, leading to potential pickup errors.

Currently, the fast pickup for parallel robots with 2D machine vision is the mainstream method [30–33]. These methods use template matching, edge contour extraction, and centroid calibration for planar centroid pickup [34]. They offer high pickup accuracy, rapid inference, and suitability for noisy industrial environments [35]. However,

most of these methods are limited to 2D pickups and lack depth information. They can only handle objects with fixed heights and cannot provide high-precision information for multiple targets in industrial environments. 3D pickup enables robots to perform various multi-target pickup work [36, 37]. Therefore, this paper conducts a 3D pickup estimation method for industrial parallel robots based on point cloud simplification and registration. The main contributions of this study are as follows:

1. A spatial domain point cloud segmentation method is proposed to separate the objects to be picked up from the robot environment and extract the incomplete point cloud information of objects.
2. The FF-WLOP method based on FPFH and WLOP algorithms is proposed to effectively separate objects that need to be picked up from the robot's environment and extract the relevant incomplete object point cloud information.
3. A pickup point estimation method based on optimal projection points is proposed to suction cup pickup of parallel robots and employ the D-SVD method for precise hand-eye calibration.
4. An experimental platform of a parallel robot is built, and the reliability of the study is verified by conducting point cloud processing of objects, hand-eye calibration of parallel robots, and 3D pickup experiments.

The structure of this paper is as follows: Sect. 2 provides a detailed explanation of the general design framework. Section 3 introduces the point cloud preprocessing, the optimal projection point strategy and the D-SVD hand-eye calibration. Section 4 carries out the related experiments. Section 5 summarizes the conclusions and outlook.

2 Overall framework design

In the majority of industrial parallel robot scenarios, a fixed single camera is commonly employed. A single view cannot get the complete point cloud of an object or determine its barycenter coordinates and pickup point coordinates. Nevertheless, by matching the complete point cloud of the same object, it is possible to perform the necessary pickup task. This paper focuses on the research of an industrial parallel robot equipped with a suction cup end-effector. To address the challenges of object positioning and pickup point evaluation, a method for point cloud simplification and registration is proposed. The overall framework is shown in Fig. 1.

The framework is divided into 7 steps. Steps 1 and 2 are data acquisition, using a 3D scanner to obtain a complete point cloud of objects to create a point cloud dataset. The camera combines the depth image and color image to generate point

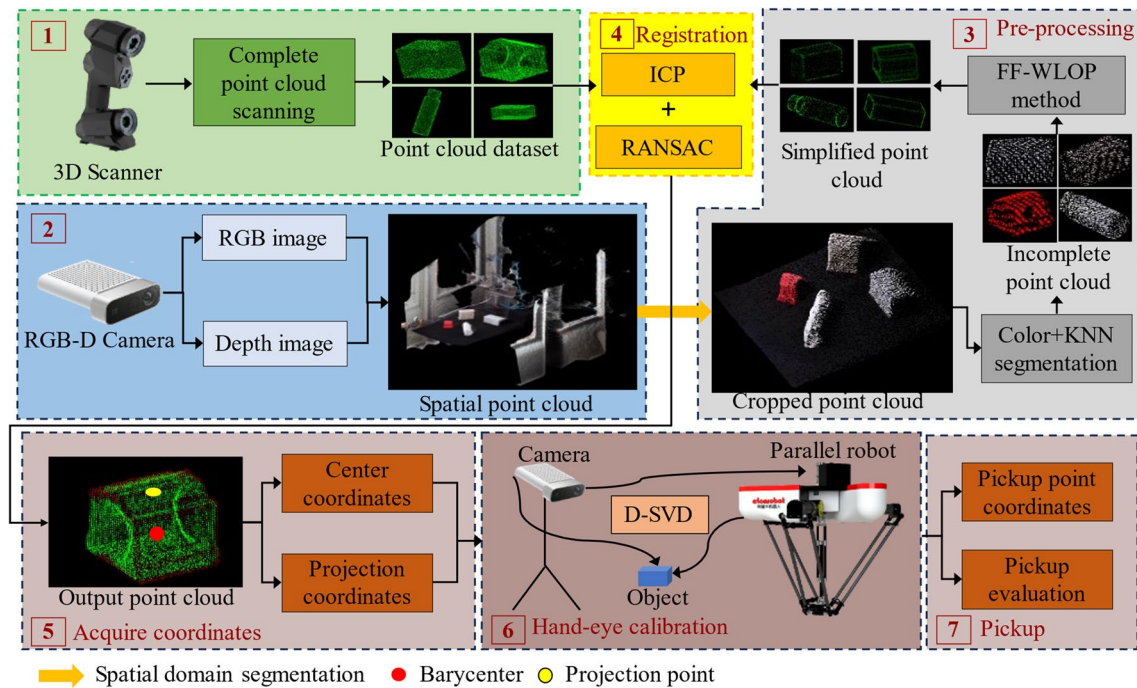


Fig. 1 Overall framework

cloud data. Step 3 is the point cloud simplification process, employing a spatial domain point cloud segmentation method to separate the background in the robotic scene. Color segmentation eliminates the bottom background and the K -nearest neighbor (KNN) algorithm is applied to segment the point cloud of each object, which is then simplified using the FF-WLOP method. Step 4 focuses on point cloud registration, the complete object point cloud information in the dataset is aligned with the simplified incomplete point cloud using Random Sample Consensus (RANSAC) and iterative closest point (ICP) algorithms to obtain the registration matrix. Step 5 involves obtaining the coordinates. The centroid coordinates of the complete point cloud are calculated to determine the object’s position, while the centroid coordinates of the incomplete point cloud are derived through the registration matrix. The optimal projection point method is utilized to determine the position of the object’s pickup point. Step 6 is hand-eye calibration, enhancing the SVD hand-eye calibration method with dynamical-weight division, and calculating the object coordinate transformation relationship within the point cloud space and the real robot space. Lastly, Step 7 is the robot pickup process.

3 Method

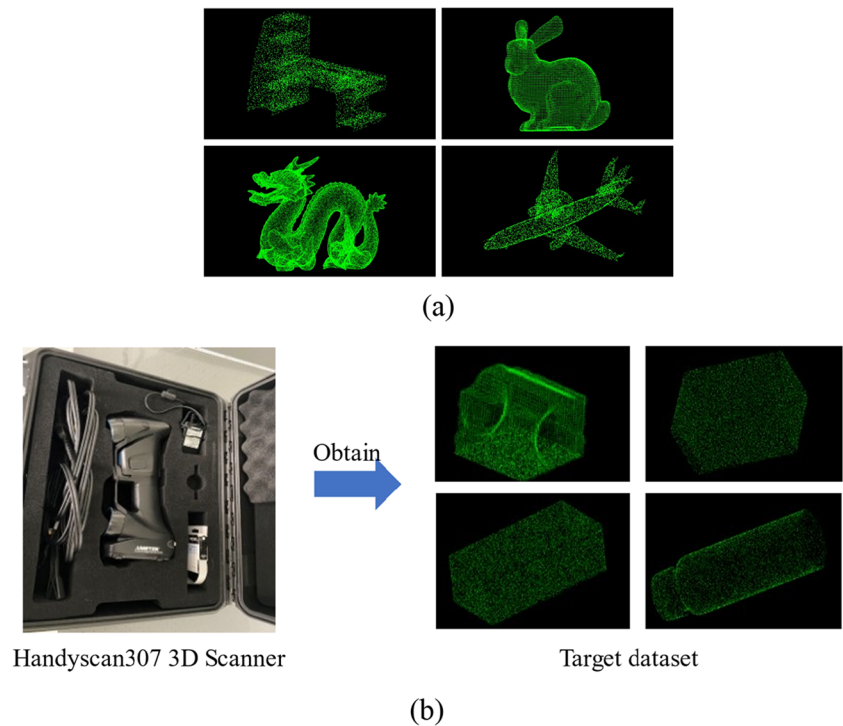
3.1 Data acquisition

Data acquisition includes the acquisition of the complete point cloud in the dataset and the point cloud data synthesized by the camera after acquiring the depth image and the RGB image.

3.1.1 Dataset preparation

A total of 100 sets of point cloud data are chosen from the ModelNet40 dataset, which is a publicly available point cloud dataset provided by the Stanford 3D Scanning Resource Library at Stanford University [38]. Twenty sets of complete point cloud data are captured from the Target dataset. A hand-held Handyscan307 laser scanner is used for point cloud acquisition. The ModelNet40 dataset is used for algorithm validation and registration evaluation, and the Target dataset is used for pickup experiments and localization evaluation. Some complete point cloud data in the datasets is shown in Fig. 2.

Fig. 2 Dataset preparation. **a** ModelNet40. **b** Target



3.1.2 Point cloud acquisition

In industrial settings, multiple devices exist around parallel robots on production lines. The interference of the background causes trouble for point cloud acquisition and processing. The results of industrial environment point cloud acquisition are shown in Fig. 3. The depth camera only needs to acquire the object point cloud to solve its barycenter coordinates for robot pickup, it is limited by the environment and inadvertent acquisition of other redundant point clouds present in the surrounding environment.

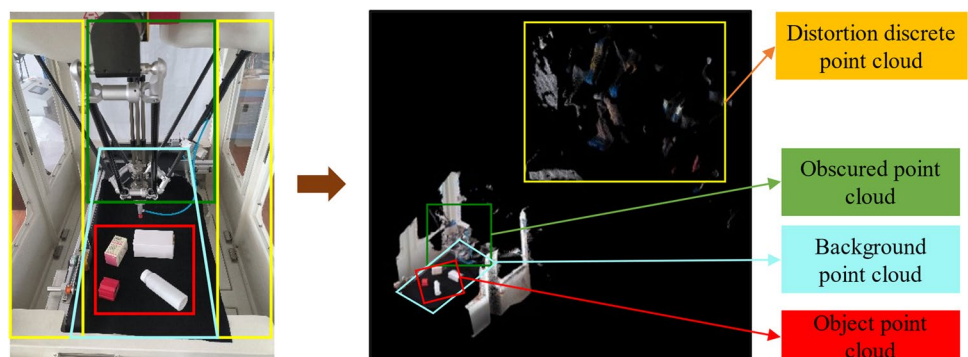
The overall point cloud, denoted as P , consists of various components including the distortion point cloud (P_D), obscured point cloud (P_R), where objects are occluded by

the robot body, the background point cloud (P_B), and the incomplete object point cloud (P_O). The extraction of P_O from P is necessary to facilitate the pickup operation.

3.2 Point cloud segmentation

To extract the object's point cloud from the surrounding environment, a segmentation process is employed in this study. The segmentation approach involves utilizing minimum enclosing box, color, and KNN segmentation to preserve the effective features of the incomplete object point cloud.

Fig. 3 Point cloud acquisition



3.2.1 Spatial domain background segmentation

Point cloud segmentation involves extracting the target object’s 6D regular stereo spatial domain from the surrounding point cloud. In this paper, a spatial domain-based background segmentation method is proposed to separate the background from the spatial domain where the object point cloud resides within the point cloud space. The specific method is shown in Fig. 4.

As shown in Fig. 4a, first obtain the limit edge point information of P , search the spatial coordinates in the point cloud, and find the limited position in the x , y , and z directions are denoted as x_{\min} , y_{\min} , and z_{\min} and x_{\max} , y_{\max} , and z_{\max} , respectively. The size of the minimum enclosing box is then calculated as follows:

$$\begin{cases} l_x = x_{\max} - x_{\min} \\ l_y = y_{\max} - y_{\min} \\ l_z = z_{\max} - z_{\min} \end{cases} \quad (1)$$

where l_x , l_y , and l_z are the length of the box edges. P_D is distant from the other point cloud and is manually divided into regions for elimination. Next, the edge limit point cloud of P_B and P_R is obtained, and the limited positions of x , y , and z are denoted as x_{\min}^1 , y_{\min}^1 , and z_{\min}^1 and x_{\max}^1 , y_{\max}^1 , and z_{\max}^1 , respectively. P_R is determined by the end-effector bottom position h , where the object is located in the spatial domain. The size of minimum enclosing box in the P_B space domain is as follows:

$$\begin{cases} l_{x2} = x_{\max}^1 - x_{\min}^1 \\ l_{y2} = y_{\max}^1 - y_{\min}^1 \\ l_{z2} = z_{\max}^1 - z_{\min}^1 - h \end{cases} \quad (2)$$

where l_{x2} , l_{y2} , and l_{z2} are the length of the box edge in P_B space domain. To mitigate any distortion in point cloud acquisition and minimize environmental disturbances, the above steps are repeated for a total of N times, and the P_R spatial domain is eliminated. The size of optimal enclosing box in the P_B spatial domain is calculated as follows:

$$\begin{cases} l_{tx} = \frac{1}{N} \sum_{n=1}^N (x_{\max}^n - x_{\min}^n) \\ l_{ty} = \frac{1}{N} \sum_{n=1}^N (y_{\max}^n - y_{\min}^n) \\ l_{tz} = \frac{1}{N} \sum_{n=1}^N (z_{\max}^n - z_{\min}^n - h) \end{cases} \quad (3)$$

where l_{tx} , l_{ty} , and l_{tz} are the optimal length of the box edge in P_B space domain. Finally, the black background at the bottom is removed by using color segmentation, while retaining the incomplete object point cloud P_O . The overall result is shown in Fig. 4b. Theoretically, the camera position remains unchanged, and point cloud segmentation only requires retaining the point cloud information in the rectangular box for subsequent object segmentation, which is an offline operation. This method is consistent with the eye-to-hand calibration layout.

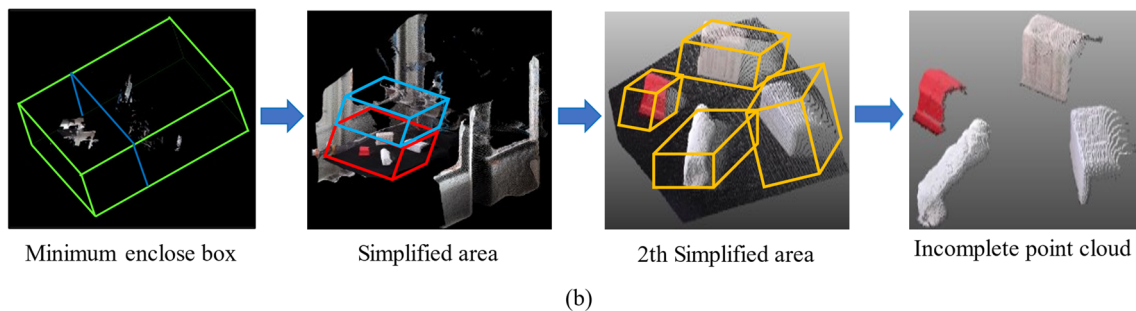
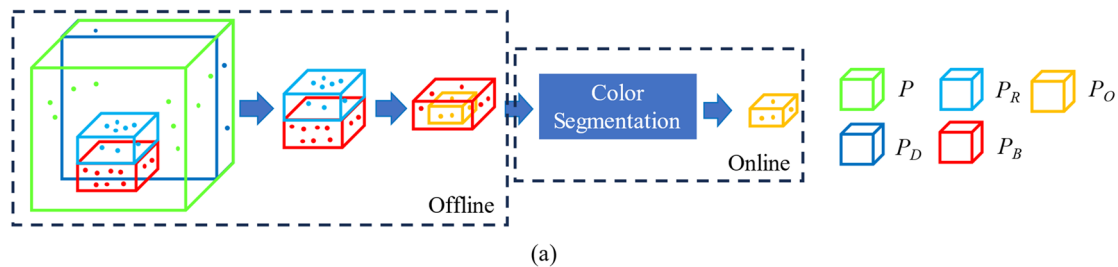


Fig. 4 Space domain segmentation. a Spatial domain segmentation process. b Actual segmentation process

3.2.2 Object segmentation

KNN algorithm computes the Euclidean distance within point cloud data. The basic principle is to select the clustered seed points P_m , define the maximum and minimum number of points in the clusters, and search the k -nearest neighboring points P_m^k within a distance P_m . The Euclidean distance between the P_m^k and P_m can be calculated as follows:

$$d(P_m, P_m^k) = \sqrt{\sum_{k=1}^n (P_m - P_m^k)^2} \tag{4}$$

where $k = \{1, 2, \dots, n\}$ until the clustered set of points no longer changes. The clustered point cloud is output, and the set of the object point cloud P_O is classified as a single-object point cloud.

3.3 Point cloud simplification

3.3.1 FPFH feature point extraction

In industrial robot applications, the environments often consist of repetitive feature points, leading to non-unique point cloud data. The FPFH method proves to be effective in extracting edge features of each object. It uses multi-dimensional histograms to capture the geometric characteristics of the surrounding k -neighborhood. By operating in a high-dimensional hyperspace, the histogram-based representation remains robust across various sampling densities and noise levels. This method offers a measurable information space for accurate feature representation.

In a spatial domain, let us define two points P_q and P_{kl} with their normal lines n_q and n_{kl} , calculate the relative difference between the points and with n_q as the u -axis, and determine the (u, v, w) spatial coordinate system by the right-hand rule, and the influential domain of FPFH is shown in Fig. 5.

The local coordinate system (u, v, w) is translated to P_{kl} , then the inter-normal deviation can be expressed by a quaternion $\langle \alpha, \theta, \varphi, d \rangle$, and the Euclidean distance between

two points is denoted as d . The transformation relation is as follows:

$$\begin{cases} \alpha = v \times n_q \\ \theta = \arctan(w \cdot n_q, u \cdot n_{k1}) \\ \varphi = u \cdot \frac{(P_{k1} - P_q)}{d} \\ d = \|P_{k1} - P_q\|_2 \end{cases} \tag{5}$$

The histogram of the relationship between the target point P_q and its immediate neighbors P_k is defined as the spherical point feature histogram (SPFH) and a quaternion $\langle \alpha, \theta, \varphi, d \rangle$ is computed between each point in the dataset and all of its immediate neighbors within a spherical region. The range of the neighborhood for each point is redefined, and the final histogram is computed by the weighted values of the neighboring. The weighted formula is as follows:

$$FPFH(P_q) = SPFH(P_q) + \frac{1}{k} \sum_{n=1}^k \frac{1}{\omega_n} \cdot SPFH(P_n) \tag{6}$$

where weight ω_n is the distance between the target point P_q and its immediate neighbors P_n . The re-weighted neighborhood influence region centered on P_q is finally obtained. The extraction effect is shown in Fig. 6, taking the captured target point cloud as an example, the green point cloud is the acquired feature points.

3.3.2 WLOP algorithm

WLOP algorithm introduces $\eta(r) = -r$ instead of $\eta(r) = 1/(3r^3)$ in the LOP algorithm as a projection repulsion factor so that it does not converge prematurely. Given a point set $P = \{p_j\}_{j \in J} \subset R^3$, define a point set $X = \{x_i\}_{i \in I} \subset R^3$ projected into P . The projected point set is denoted as $Q = \{q_i\}_{i \in I}$, and Q to satisfy the minimum of the sum of weighted distances from the projected points to P . The current iteration is denoted as $X^k (k = 0, 1, \dots, n)$, and the iterative minimization function is as follows:

Fig. 5 FPFH area in relationship to the coordinate system

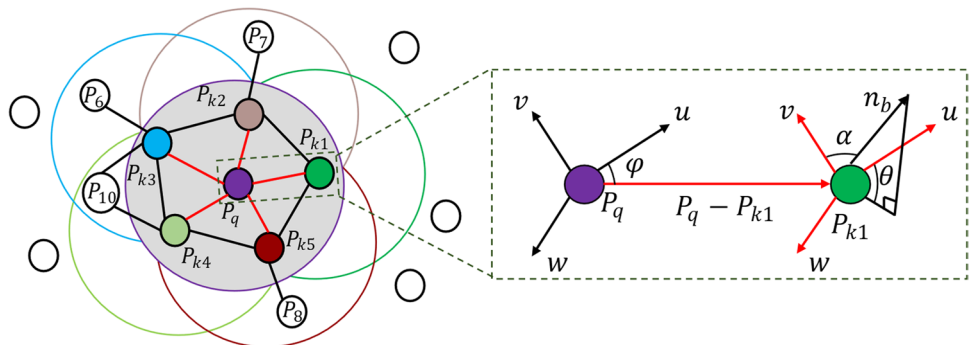
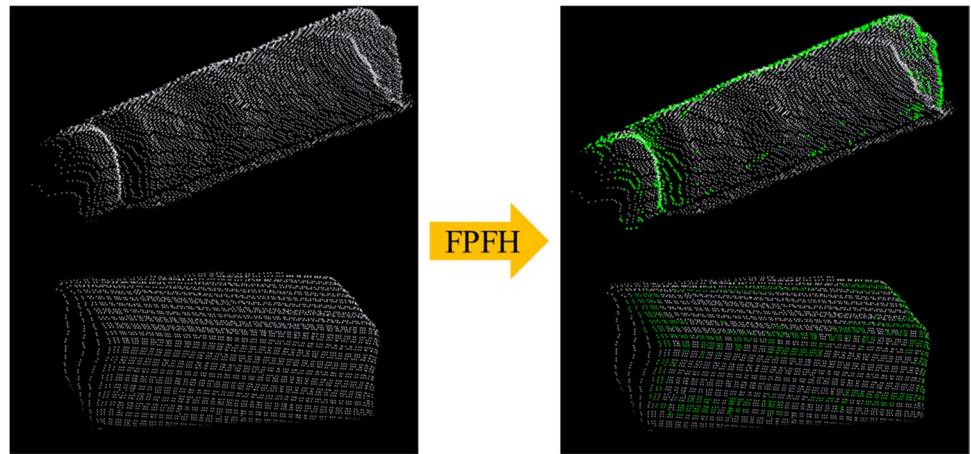


Fig. 6 Feature extraction effect



$$\sum_{i \in I} \sum_{j \in J} \|x_i - p_j\| \theta(\|\xi_{ij}^k\|) / v_j + \mu \sum_{i \in I \setminus \{i\}} \eta(\|x_i - x_{i'}^k\|) \theta(\|\delta_{i i'}^k\|) w_i^k \quad (7)$$

where $\xi_{ij}^k = x_i^k - p_j$, $\delta_{i i'}^k = x_i^k - x_{i'}^k$, and $\theta(r) = e^{-r^2 m / d_{bb}}$ are the smooth weight functions, d_{bb} is the diagonal length of the input point cloud boundary, m is the number of points, the balance term is $I\{i\}$, the size is controlled using a repulsion factor μ , and $\eta(\cdot)$ is the projected repulsion factor. $\|\cdot\|$ is 2-norm.

The optimization criterion of the LOP algorithm is related to the number of paradigms. When the input point cloud exhibits a highly irregular distribution, the algorithm follows this trend, often resulting in a large number of redundant points in the feature region of the point cloud model. Therefore, the WLOP algorithm defines local density weights based on LOP. These weights strengthen the repulsion between points in dense regions, particularly where the point cloud is denser

at the sharp features of the object. Local density weights v_j and w_i^k are introduced at input point cloud P and X^k is as follows:

$$v_j = 1 + \sum_{j' \in J \setminus \{j\}} \theta(p_j - p_{j'}) \quad (8)$$

$$w_i^k = 1 + \sum_{i' \in I \setminus \{i\}} \theta(\delta_{i i'}^k) \quad (9)$$

The position of the projection x_i^{k+1} has been updated to the following:

$$x_i^{k+1} = \sum_{j \in J} p_j \frac{\alpha_{ij}^k / v_j}{\sum_{j \in J} \alpha_{ij}^k / v_j} + \mu \sum_{i' \in I \setminus \{i\}} \delta_{i i'}^k \frac{w_{i'}^k \beta_{i i'}^k}{\sum_{i' \in I \setminus \{i\}} w_{i'}^k \beta_{i i'}^k} \quad (10)$$

where $\alpha_{ij}^k = \frac{\theta(\|\xi_{ij}^k\|)}{\|\xi_{ij}^k\|}$ and $\beta_{i i'}^k = \frac{\theta(\|\delta_{i i'}^k\|) |\eta(\|\delta_{i i'}^k\|)|}{\|\delta_{i i'}^k\|}$. Optimized point cloud clusters remove redundant and discrete points and produce dense point clouds in obvious feature positions.

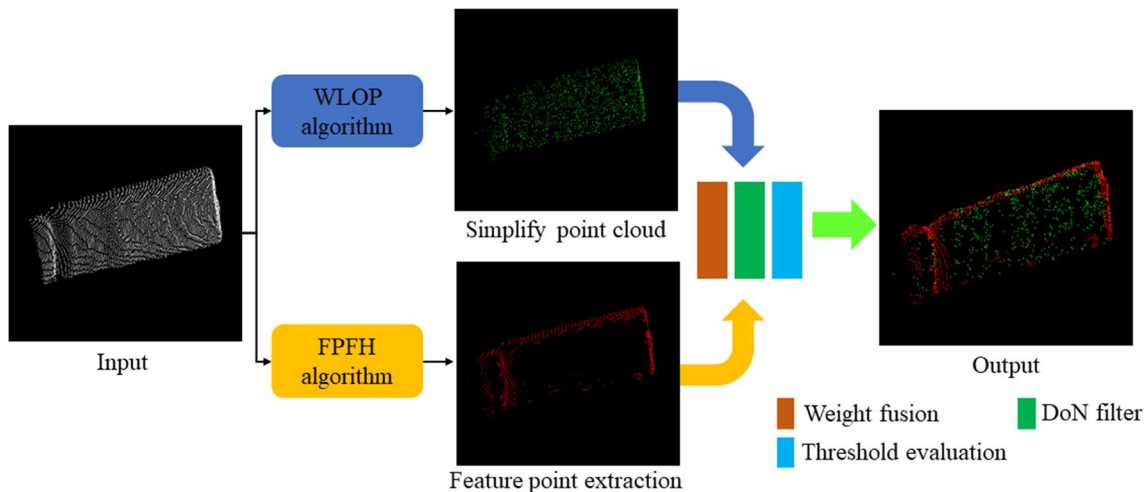


Fig. 7 Point cloud fusion

3.3.3 FF-WLOP

The WLOP algorithm is capable of simplifying point cloud data by achieving uniformity. However, it often results in a poor object feature extraction. The FPFH algorithm is effective in extracting feature points, but may lead to point cloud distortion and loss of uniform distribution. Therefore, a point cloud fusion method named FF-WLOP is proposed, which combines the optimization strengths of both algorithms. The specific idea of the FF-WLOP method is shown in Fig. 7.

The process of fusing the target point cloud is shown in Fig. 7. The WLOP algorithm is applied to simplify the bottle point cloud with some distortion in the bottle mouth features. Subsequently, the FPFH algorithm is utilized to extract the key feature points from the point cloud. Once the point cloud fusion is completed, the resulting target point cloud is characterized by a high degree of simplification and minimal distortion. The FF-WLOP method overlays the advantages of the two methods, and the following is the hybrid method strategy process.

Firstly, the point cloud data of FPFH and WLOP algorithms need to be weighted and fused, and the feature point cloud of FPFH is adjusted to focus on the selection of the external contour and increase the weight of the edge environment exploration in the neighborhood of the target point P_{qt} , and the set of the edge points is P_e , and the weighted improvement formula of FPFH is as follows:

$$\text{FPFH}(P_{qt}) = \text{SPFH}(P_{qt}) + \frac{1}{n} \sum_{k=1}^n \left(\frac{\alpha}{\omega_k} \cdot \text{SPFH}(P_i) + \frac{\beta}{\omega_k} \cdot \text{SPFH}(P_e) \right) \quad (11)$$

where α , β are the weighted factors, which affect the acquisition of edge feature points. Weight ω_k is the distance between the target point P_{qt} and P_i , P_e . To pick up complex objects, the robot needs to evaluate the surface features, whereas for the parallel robot primarily considers weighing edge points. The internal feature points are further eliminated to reduce both the amount of point cloud data and the inference time.

By fusing point cloud data with identical space coordinates, overlapping points of objects are eliminated.

Then, difference of normal (DoN) filtering is performed to remove the fused noise points. Finally, a dynamic evaluation threshold strategy is performed to evaluate different incomplete object point cloud P_o simplification results. The data volume of the incomplete point cloud P_o is denoted as D_o , the data volume after point cloud fusion is denoted as D_w , the data volume after DoN filtering is denoted as D_f , and the final simplified data volume is denoted as D_s . The threshold evaluation process is shown in Fig. 8.

The edge discrete spatial point distances present in D_f are calculated, and if Euclidean distance $d(p_{qt}, p_e) < \varepsilon$, then the point cloud point count is performed; conversely, the filtering calculation is performed again. When $D_f < \frac{D_o}{2}$, the final simplified data D_s is obtained; conversely, when $D_f < \frac{3D_o}{4}$, statistical filtering is followed by DoN filtering, the condition does not hold then the point cloud exists a large noise and redundant point set, again weighted FPFH operation. This process improves the quality of the subsequent registration by focusing on objects with significant geometric distortion. The aim of the threshold evaluation is to reduce the data volume of the incomplete point cloud of the object by more than half, and to eliminate scattered points with substantial distortion through multiple filtering.

3.4 Point cloud registration

Point cloud registration can be divided into coarse and fine registration. In this study, the random sample consensus the RANSAC algorithm and ICP algorithm are employed for coarse and fine registration, respectively. The RANSAC-based coarse registration algorithm does not require any specific initial positions for both the source and target point clouds, and it can still achieve more accurate registration results even when there is a low overlap between the two sets of point cloud data. Before the registration process, the source and target point clouds undergo

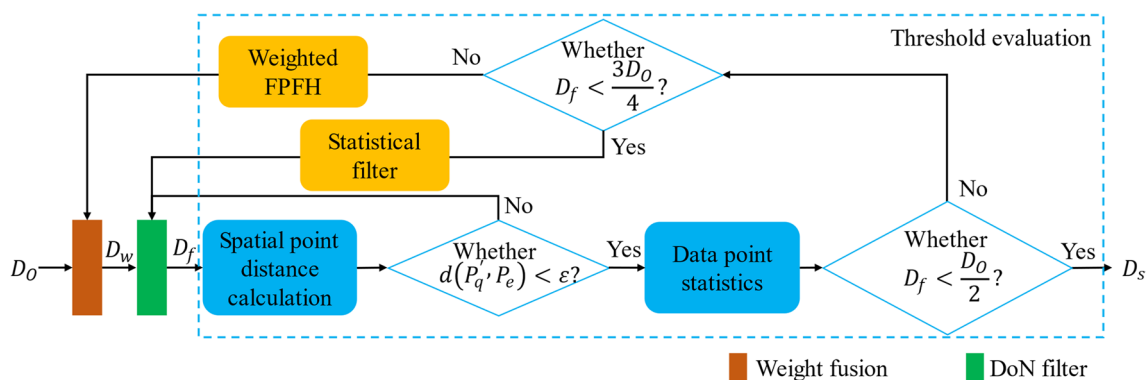


Fig. 8 Threshold evaluation

preprocessing operations, with a higher overlap between the two groups of point clouds, the better the registration effect. The source point cloud is solved by the RANSAC algorithm to obtain the coarse registration matrix H_R so that the overlap between the source and target point clouds is improved.

The fine registration of point cloud using the ICP algorithm involves initially selecting registration element points and then carries out the determination of the inter-point set mapping matrix H_I by obtaining optimal matching parameters and a coordinate transformation matrix, while adhering to certain constraints. The transformation matrix obtained from the registration is as follows:

$$H_{RI} = H_I H_R \quad (12)$$

where $H_{RI} \in \mathbb{R}^{4 \times 4}$, the point cloud registration can convert the complete point cloud coordinates of the object under the 3D scanner by H_{RI} to the incomplete point cloud coordinates of the object under the camera coordinate system. The registration result is shown in Fig. 9.

Figure 9 depicts the process of aligning the bottle and box. The red point cloud represents the incomplete target point cloud after FF-WLOP simplification. The green point cloud represents the complete point cloud from the Target dataset. By applying the alignment algorithm, a transformation matrix of high accuracy is obtained.

3.5 Robot pickup positioning

3.5.1 Pickup process

The essence of the pickup positioning and calibration process is to convert the object voxel coordinates $(x_c, y_c, \text{and } z_c)$ in the point cloud space to the actual distance space coordinates (x_w, y_w, z_w) in the robot space. As shown in Fig. 10, the point cloud data acquired by the depth camera can calculate voxel coordinates of object. Through the combined use of RANSAC and ICP registration, along with the optimal projection point method, the pickup and calibration of the parallel robot are accomplished. The calibration process is an online operation, which is carried out in real-time. The point cloud pickup point judgment is an offline operation, which needs to be completed in advance.

3.5.2 Barycenter calculation

The camera-captured incomplete point cloud of an object captured does not yield accurate barycenter points, but it is aligned using RANSAC and ICP algorithms. The solution of the barycenter coordinates for incomplete point clouds can be converted to solving the barycenter coordinates for complete point clouds from the same objects in the database. The formula for calculating the barycenter of a point cloud is as follows:

Fig. 9 Registration results: **a** bottle and **b** box

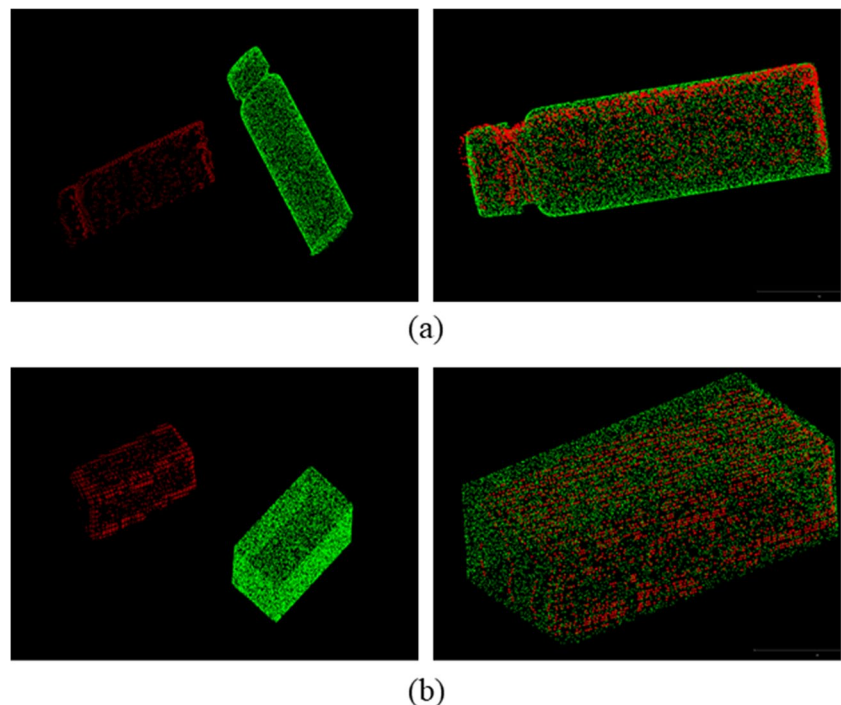
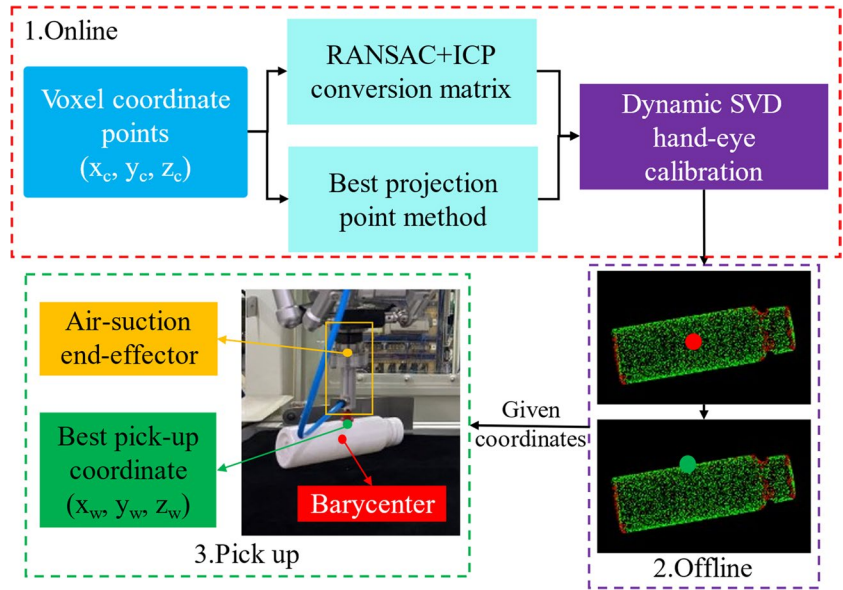


Fig. 10 Pickup process



$$P_c = \frac{1}{n} \sum_{i=0}^n m_i r_i \tag{13}$$

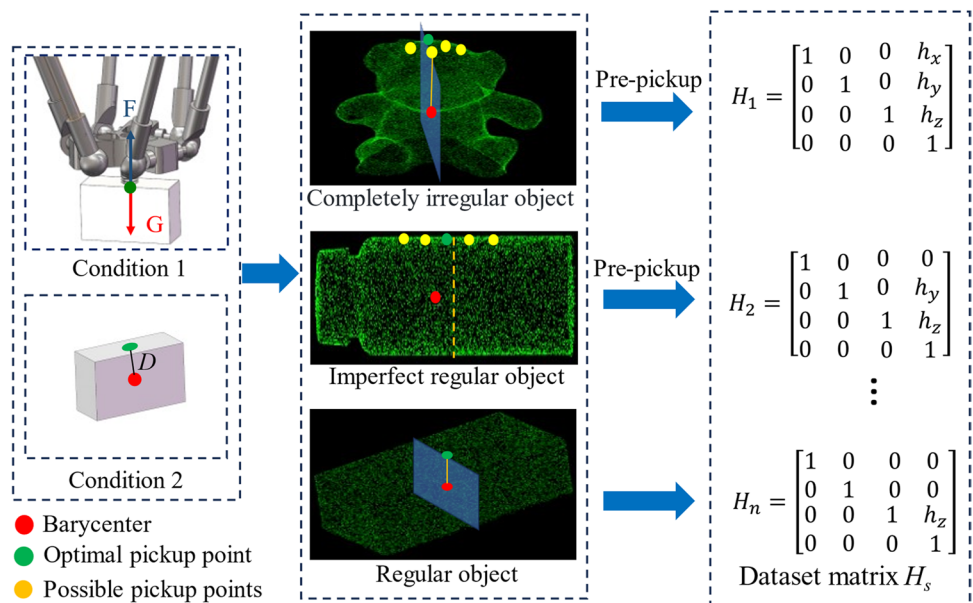
where n is the total point cloud data volume, $r_i = (x_i, y_i, z_i), i = 0, 1, 2, \dots, n$ and m_i is the mass of a single point. When calculating the barycenter of the point cloud, let $m_i = 1$, then the formula for the barycenter coordinates is as follows:

$$P_c = \frac{1}{n} \left(\sum_{i=0}^n x_i, \sum_{i=0}^n y_i, \sum_{i=0}^n z_i \right) \tag{14}$$

3.5.3 Optimal projection point strategy

The suction cup end-effector is typically required to be attached to the upper surface of the objects. The quality of the pickup is assessed based on two conditions: firstly, the adsorption seal is examined, ensuring that the adsorption force F balance the object’s gravity G to maintain stability; secondly, the distance from the barycenter is considered, where a closer proximity between the pickup point and the barycenter D indicates more reliable adsorption. The closer the pickup point is to the barycenter of the object, the smoother the pickup process becomes, resulting in a

Fig. 11 Selection of the optimal projection point



more stable and controlled operation. The optimal projection point strategy is shown in Fig. 11.

As shown in Fig. 11, condition 1 ensures successful object pickup, while condition 2 enables precise positioning. General objects are categorized as regular, imperfectly regular, and completely irregular. The red point is the barycenter, and the green point is the optimal pickup point. The yellow points represent possible pickup points, but a pre-pickup action is necessary to determine if they are indeed suitable pickup locations.

For completely irregular objects, attempts need to be made to pick them up and evaluate the degree of adsorption by the end-effector. In situations where the barycenter and the uppermost point are not co-planar and cannot be adsorbed, the focus shifts to finding a more intact object on the upper surface for pickup attempts. Therefore, it is necessary to manually assign a specific transformation matrix H_1 .

For imperfect regular and regular objects, the optimal pickup point coincides with the barycenter point in the same plane. To imperfectly regular objects, it is necessary to perform a pre-pickup operation on the upper surface in the plane where the barycenter is located, and obtain the transformation matrix H_2 between barycenter and the pickup point. To regular objects, there is no need for a pre-pickup operation. The optimal pickup point is directly above the barycenter, and only the distance in the z -direction needs to be determined, which is determined by the object's height, the transformation matrix is H_n .

The set of transformation matrices for the barycenter and the best pickup point for all objects is $H_s = \{H_1, H_2 \dots, H_n\}, s \leq n$. While it completes the solution of the barycenter of a specific target, it also extracts its unique transformation matrix H_s to automatically convert

the barycenter coordinates of the dataset to the object pickup coordinates in the camera's coordinate system.

3.6 Robot hand-eye calibration

3.6.1 Eye-to-hand calibration model

The hand-eye calibration is based on the relationship between the relative positions of the depth camera, parallel robot, and object to solve the coordinate transformation relationship of the object under the point cloud space and the robot space as shown in Fig. 12. Define transformation matrix for the target object in the camera coordinate system as H_o^c , the corresponding target transformation matrix in the robot world coordinate system as H_o^w , and the transformation matrix of the spatial point cloud coordinates in the camera coordinate system to the real coordinates in the robot world coordinate system as H_c^w . Object point cloud voxel coordinates are transformed in the robotic world coordinate system with the relationship is as follows:

$$H_o^w = H_c^w H_o^c \tag{15}$$

where $H_o^c = H_s H_{RI}$, $H_o^c = \begin{bmatrix} R^{3 \times 3} & t^{3 \times 1} \\ 0 & 1 \end{bmatrix}$, R is the rotation matrix, and t is the translation matrix. H_{RI} is the dynamic matrix, which changes with the object's position. H_s, H_c^w are the static matrix, H_s need to be set in advance, and H_c^w can be solved once.

3.6.2 D-SVD hand-eye calibration

The SVD method utilizes the end-effector center to calibrate the significant point set of the object. It is performed through multiple poses in both point cloud space and

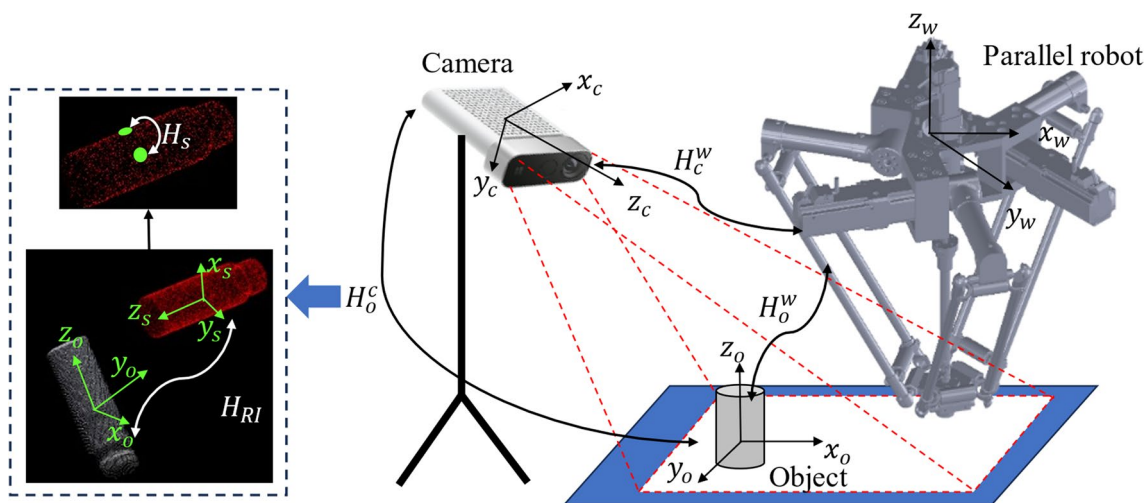


Fig. 12 Calibration model

robot workspace. The hand-eye calibration is disturbed by the point cloud acquisition, which has a large impact on the calibration accuracy. Hence, for point cloud calibration, the SVD hand-eye method with dynamic weights is employed to calculate the transformation matrix from the camera coordinate system to the robot world coordinate system. The fundamental steps of this method are as follows:

The point set in the point cloud space is denoted as $P^c = \{p_1^c, p_2^c, \dots, p_n^c\}$. The point set in the robot space is denoted as $P^w = \{p_1^w, p_2^w, \dots, p_n^w\}$. The point set relationship obtained by the least squares method is denoted as follows:

$$f(R, t) = \operatorname{argmin} \sum_{i=1}^n \omega_i \| (Rp_i^c + t) - p_i^w \|^2 \quad (16)$$

where $\omega_i = 1$ is the point weight corresponding to the point set. The center of gravity of point sets P^c and P^w are denoted as \hat{P}^c and \hat{P}^w :

$$\begin{cases} \hat{p}^c = \sum_{i=1}^n \omega_i p_i^c / \sum_{i=1}^n \omega_i \\ \hat{p}^w = \sum_{i=1}^n \omega_i p_i^w / \sum_{i=1}^n \omega_i \end{cases} \quad (17)$$

The center of gravity from each point in the point set obtains a new point set x_i with y_i are denoted as follows:

$$\begin{cases} x_i = p_i - \hat{p}^c \\ y_i = p_i - \hat{p}^w \end{cases} \quad (18)$$

It can be solved from Eqs. (16) and (18):

$$f(R, t) = \operatorname{argmin} \left(\sum_{i=1}^n \omega_i x_i^T x_i - 2 \sum_{i=1}^n \omega_i y_i^T R x_i + \sum_{i=1}^n \omega_i y_i^T y_i \right) \quad (19)$$

where $\sum_{i=1}^n \omega_i x_i^T x_i$ and $\sum_{i=1}^n \omega_i y_i^T y_i$ are independent of R , $f(R, t) = \operatorname{argmin} (2 \sum_{i=1}^n \omega_i y_i^T R x_i)$. Reconstruct the center of gravity matrices X , Y and the weight diagonal matrix W is as follows:

$$X = [X_1 \ X_2 \ \dots \ X_n] \quad (20)$$

$$Y = [Y_1 \ Y_2 \ \dots \ Y_n] \quad (21)$$

$$W = \begin{bmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \omega_n \end{bmatrix} \quad (22)$$

A singular value solution for the matrix can be obtained:

$$XWY^T = U \sum V^T \quad (23)$$

The conversion relation between R and t is further found to be as follows:

$$\begin{cases} R = VU^T \\ t = \hat{q} - VU^T \hat{p} \end{cases} \quad (24)$$

H_c^w can be determined from R and t . However, when ω_i is fixed, the point set relationship will cause the calibration error to increase, so dynamic weight ω_d^i is proposed for weighting. The weighted principle is that the larger the point set error, the smaller the weight. The calibration errors ε_d^i after initial SVG calibration is as follows:

$$\varepsilon_d^i = \|(Rp_i^c + t) - p_i^w\|^2, i = 1, 2, \dots, n \quad (25)$$

Define the average of all the combined calibration errors as $\bar{\varepsilon}_d$, and the dynamic weights ω_d^i is as follows:

$$\omega_d^i = \begin{cases} 1 \varepsilon_d^i \leq \bar{\varepsilon}_d \\ e^{-|\varepsilon_d^i - \bar{\varepsilon}_d| / \varepsilon_d^i} \bar{\varepsilon}_d < \varepsilon_d^i \leq 10 \\ 0 \varepsilon_d^i > 10 \end{cases} \quad (26)$$

when $\varepsilon_d^i \leq \bar{\varepsilon}_d$, the error is within the controllable range, the initial weight $\omega_i = 1$ is used; when $\bar{\varepsilon}_d < \varepsilon_d^i < 10$, the error is large, the dynamic weights can be corrected; and when $\varepsilon_d^i > 10$, the error is too large, then the weight cannot be used for the calculation of $f(R, t)$, and it is necessary to re-collect the other points in the point set to be analyzed. Finally, the point cloud point sets are calibrated with the object point sets in the robot space. For a total of ten iterations, the dynamic weights ω_d^i are brought into Eq. (16) to complete the hand-eye calibration. There are many discrete point sets in the incomplete point cloud of the object, and the dynamic weights can better avoid the calibration error caused by the discrete points in the point cloud.

4 Experiment and results

This study includes several experiments to evaluate and analyze the effect of the FF-WLOP algorithm on point cloud registration results, positioning error in hand-eye calibration, and localization results and pickup efficiency in a robot pickup scenario. A point cloud processing experiment is conducted to verify the impact of the FF-WLOP algorithm. A hand-eye calibration experiment is performed to analyze the positioning error. Lastly, a robot pickup experiment is conducted to analyze the localization results and pickup efficiency.

4.1 Experimental preparation

The experimental setup consists of a parallel robot, an RGB-D depth camera, a demonstrator, objects for pickup, and a test computer, as shown in Fig. 13. The robot is an Atom-Delta parallel robot to pick up the objects and place

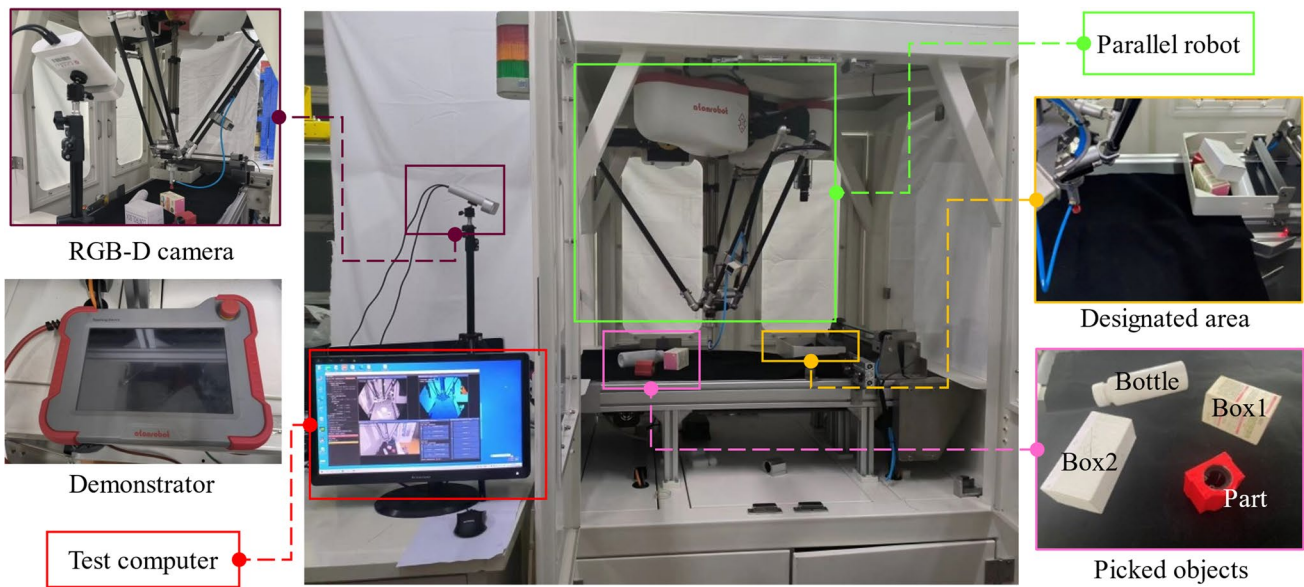


Fig. 13 Experimental platform

them in a designated area. The demonstrator is responsible for acquiring the robot's position and estimating its posture. The chosen RGB-D camera is the Microsoft Azure Kinect v3, which enables depth and color image acquisition, and a bracket is used to fix it directly in front of the robot's pickup space. The objects to be picked up include four different types: a part, a bottle, box1, and box2. Box1 and box2 vary in the smoothness of their surfaces, allowing for the evaluation of the adsorption capability and pickup stability. The test computer is utilized for point cloud pre-processing and object pickup points. It is equipped with a 12th-Gen Intel (R) Core (TM) i7-12700-CPU@2.10 GHz, Nvidia GTX3060 GPU (8G), and runs on PCL1.9.1 and Visual Studio 2017 as a point cloud environment.

4.2 Point cloud processing results and analysis

The depth camera combines depth image and RGB image to generate the point cloud data. Initially, a spatial domain segmentation is performed to determine the minimum enclosing box that encompasses the object. The region is segmented by the yellow cube, which eliminates the redundant robot's environment point cloud and retains the spatial point cloud where the object is located. Then, the background point cloud is eliminated by color segmentation, and the segmentation result is shown in Fig. 14.

The remaining point cloud is segmented by the KNN algorithm for each object. The segmented objects that are retained are shown in Fig. 15. In Fig. 15a, the incomplete point cloud segmentation result of the objects is

Fig. 14 Background segmentation result. **a** Scene space domain segmentation. **b** Background segmentation

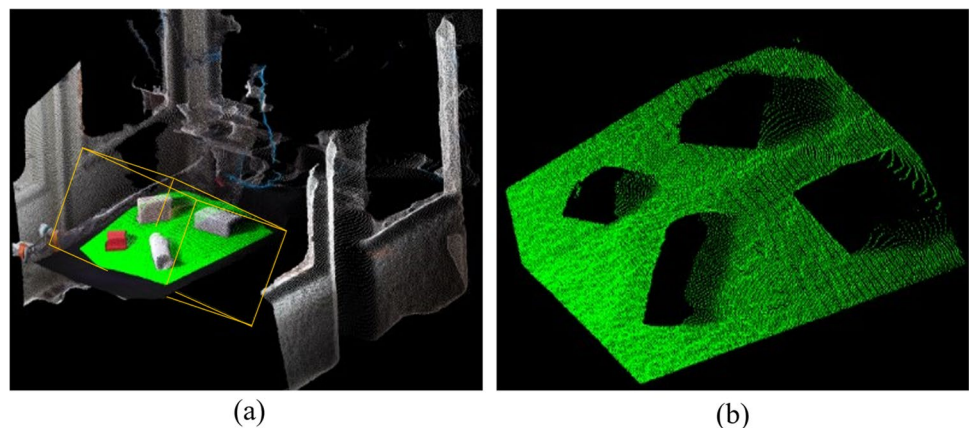
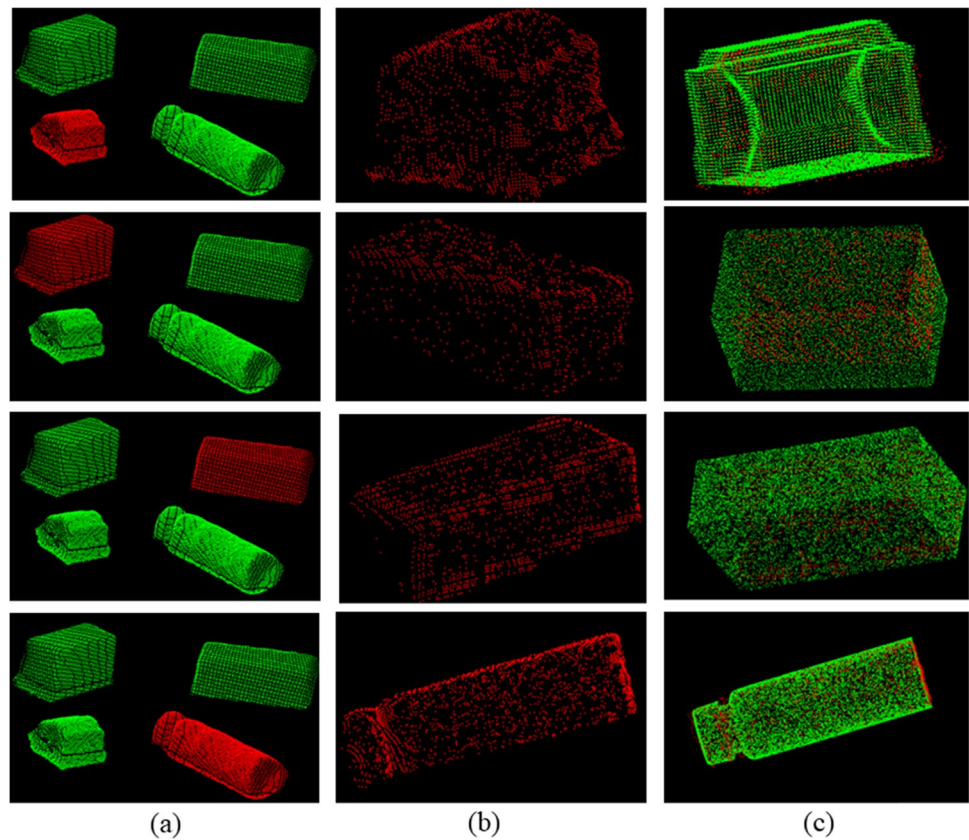


Fig. 15 Processing results. **a** Object segmentation. **b** FF-WLOP simplification. **c** Registration results



displayed, which retains the basic features, does not produce large-scale distortion, and the segmentation of a single segmented object is accurate. Figure 15b shows the simplified result of the incomplete point cloud through the FF-WLOP algorithm, which retains the basic edge contour result without losing the original features. Figure 15c shows the registration result of the simplified result with the complete point cloud from the Target dataset. The edge features of the red simplified point cloud registration well with the complete point cloud.

After the registration is completed, the transform matrix obtained from the current RANSAC and ICP registration results is shown in Table 1.

Registration work was performed, one using the point cloud data in ModelNet40 and the other using the Target dataset and the simplified point cloud. The results of the associated algorithms are evaluated. The analysis of registration errors is performed using original point clouds, and simplified point clouds of FPFH, WLOP, and FF-WLOP algorithms. The comparison results are shown in Fig. 16.

As Fig. 16 shows, the registration errors in the x -, y -, and z -axes, are smaller when using ModelNet40, ranging between -0.25 and 0.25 mm. The registration of simplified and complete point clouds exhibits larger errors due to the equipment limitations, but they still remain within a normal range. The registration error in the z direction is

larger, ranging between -0.61 and 0.64 mm. When comparing the registration errors of the simplified and complete point clouds, FPFH, WLOP, and FF-WLOP simplified point cloud, the original point cloud displays initial distortion and discrete points, resulting in poor registration accuracy. The FPFH simplification reduces the overall point cloud information and decreases the registration

Table 1 The results of registration

Target types	Transformation matrix H_{RI}
Part	$\begin{bmatrix} 0.900 & 0.426 & -0.101 & 56.848 \\ -0.043 & -0.145 & -0.989 & 696.264 \\ -0.435 & 0.894 & -0.113 & 353.933 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Bottle	$\begin{bmatrix} 0.499 & 0.568 & 0.763 & -572.895 \\ 0.310 & 0.212 & -0.365 & 332.462 \\ 0.142 & -0.184 & 0.654 & 526.055 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Box1	$\begin{bmatrix} -0.160 & -0.235 & -0.338 & -217.257 \\ 0.577 & 0.053 & -0.520 & 129.001 \\ -0.653 & 0.828 & -0.213 & 381.272 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Box2	$\begin{bmatrix} -0.038 & 0.723 & -0.700 & 89.237 \\ -0.065 & 0.212 & 0.768 & 623.734 \\ 0.959 & 0.084 & 0.437 & 243.380 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

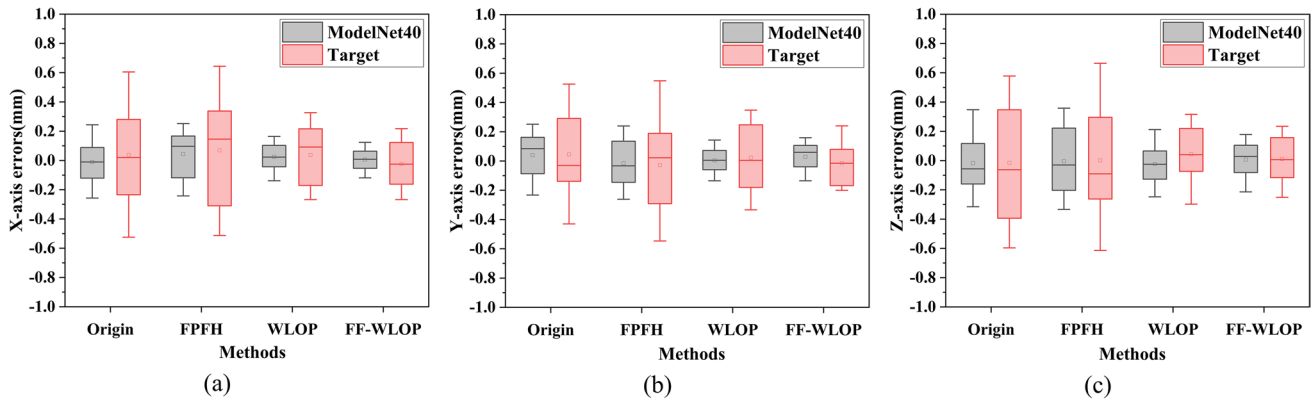


Fig. 16 Registration error analysis results. a X-axis. b Y-axis. c Z-axis

Table 2 Comparison results of registration

Method	Simplified rate	Accuracy (%)	Inference time (s)	Errors(mm)
Origin	-	93.8%	2.326	0.68
FPFH	57.3%	94.6%	2.213	0.69
WLOP	32.8%	98.5%	2.297	0.43
FF-WLOP	50.2%	99.2%	2.806	0.38

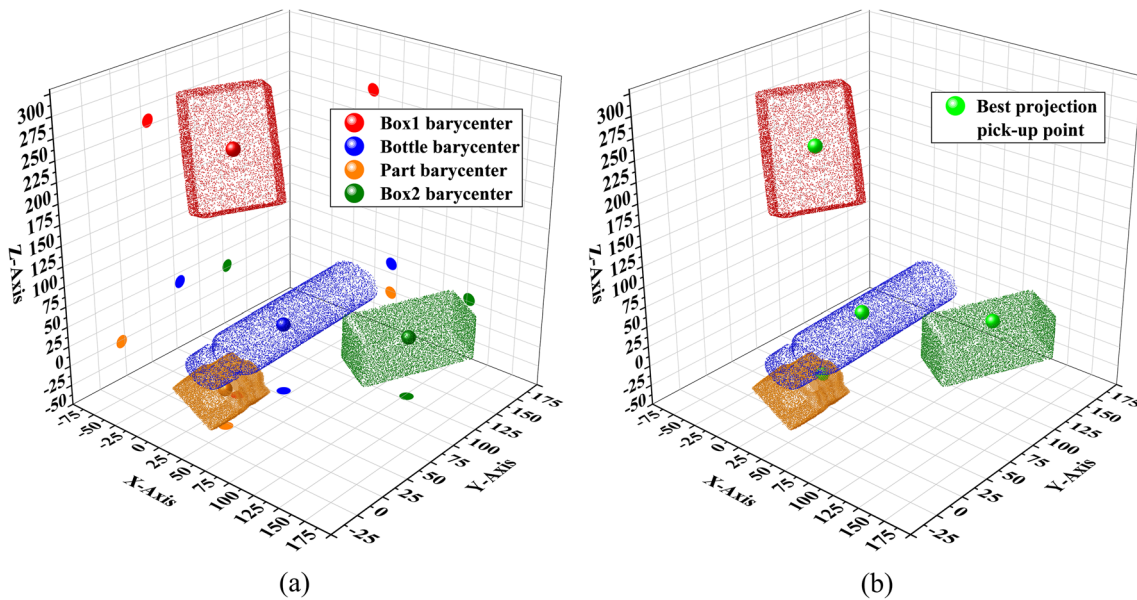


Fig. 17 Point cloud calculation results. a Barycenter calculation. b Pickup position calculation

accuracy. The registration error of the FF-WLOP method is lower than the WLOP method. The registration accuracy result is shown in Table 2.

Table 2 shows the statistical results, the registration accuracy is above 93%. The initial point cloud registration exhibits the poorest performance, while the FF-WLOP method achieves the optimal results. The

inference time refers to the duration required to align the simplified incomplete point cloud with the complete point cloud within the Target dataset. The FF-WLOP algorithm, although 0.509 s slower than the WLOP algorithm, demonstrates a higher error accuracy improvement of 0.05 mm. After the registration is completed, the barycenter of the complete point cloud in the Target

dataset is computed. The 3D coordinates of the points are picked up by the solution of the optimal projection strategy as shown in Fig. 17.

Figure 17a shows the result of barycenter acquisition, enabling the accurate spatial coordinates of the object

from the complete point cloud in the dataset. The optimal projection pickup point in Fig. 17b is located on the upper surface of the object, and the coordinates of the point cloud pickup point are obtained based on the scanner coordinate system.

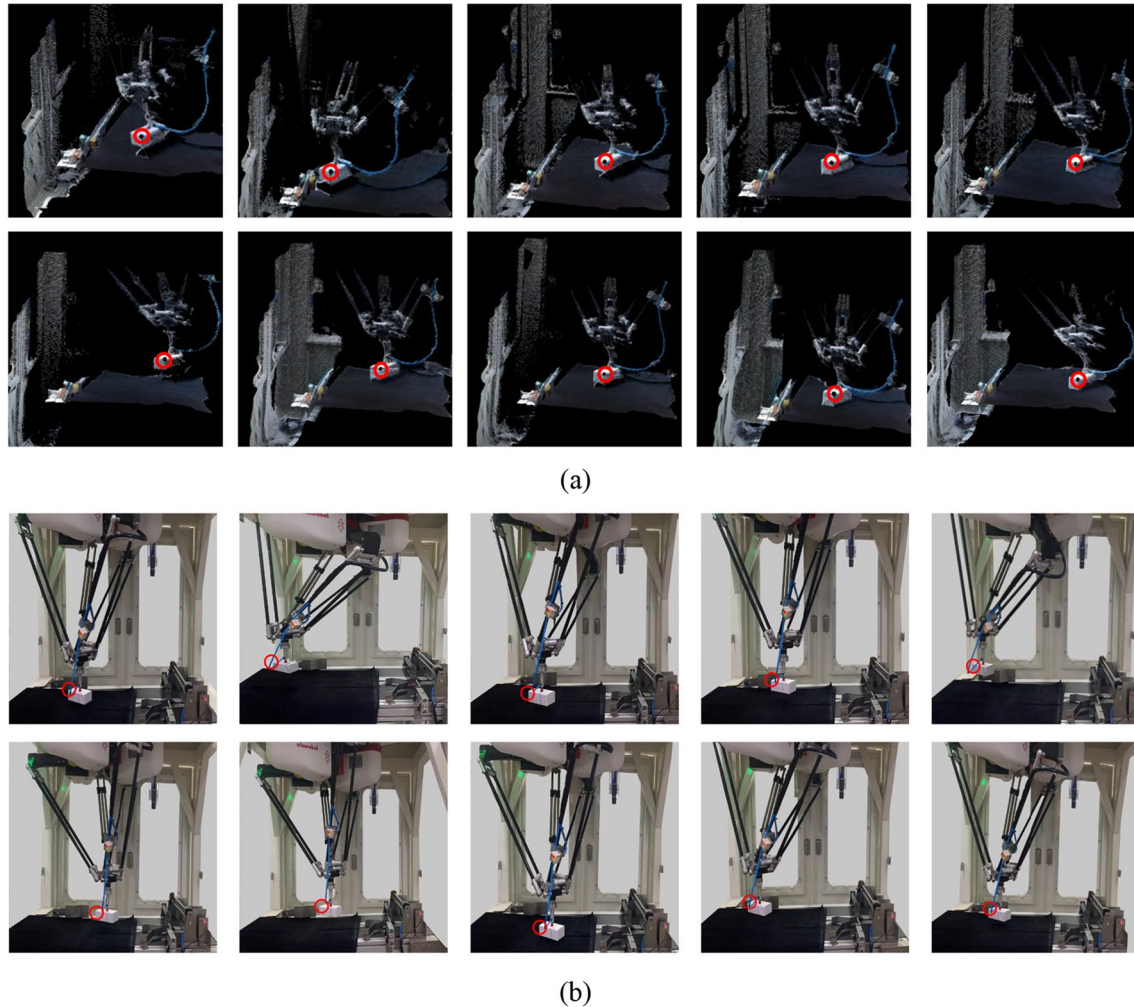


Fig. 18 Hand-eye calibration process. **a** In the point cloud environment. **b** In the robotic environment

Table 3 Hand-eye calibration experiment data

Number	Eye-x	Eye-y	Eye-z	Hand-x	Hand-y	Hand-z
1	87.064	239.932	780.102	254.139	-16.272	-537.594
2	97.131	124.028	875.893	107.339	-1.027	-506.377
3	136.962	139.141	840.015	145.15	39.555	-504.204
4	141.879	164.912	860.213	145.15	39.555	-536.975
5	162.046	172.184	816.024	184.697	62.845	-520.429
6	197.857	131.058	826.148	151.366	102.022	-500.141
7	39.107	92.417	929.208	47.04	-58.326	-502.792
8	65.101	96.126	940.961	39.221	-35.82	-517.972
9	143.753	66.054	965.302	2.078	44.025	-523.544
10	93.267	83.067	934.310	36.727	-4.924	-509.726

4.3 Calibration results and analysis

The Delta parallel robot utilizes a suction cup attached to the center point of the upper surface of an object for pickup operations. An unobstructed corner point of the target object is selected as the calibration point. The coordinates of this point are recorded under both the camera and robot coordinate system. To establish a point set capable of transforming between the two 3D spaces, the robot end-effector undergoes spatial attitude adjustment. The coordinates of the point set in the camera coordinate system can be directly obtained from the object’s point cloud. Then, the robot end coordinates are obtained by the demonstrator, and the position relationship between the corner point and the center point of the end-effector is measured to obtain the coordinate value of the point set in the world coordinate system of the robot. The D-SVD method is employed to solve the relationship between

multiple point sets, thereby determining the transformation matrix between different coordinate systems. The hand-eye calibration process is shown in Fig. 18.

A total of 10 groups of calibrated point sets is completed according to the above steps, and the two inter-mapped point sets are shown in Table 3.

The calibration matrix obtained by the D-SVD calibration is as follows:

$$H_c^w = \begin{bmatrix} 0.0005 & 0.6098 & -0.7925 & 726.3300 \\ 0.9834 & -0.1441 & 0.1102 & 18.1956 \\ -0.1815 & -0.7793 & -0.5998 & 133.0395 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

The coordinates of the pickup point in the robot coordinate system can be obtained. The analysis of the calibration results for both the original method and the D-SVD method are analyzed as shown in Fig. 19.

Fig. 19 Calibration error results. **a** Translation errors. **b** Rotation errors

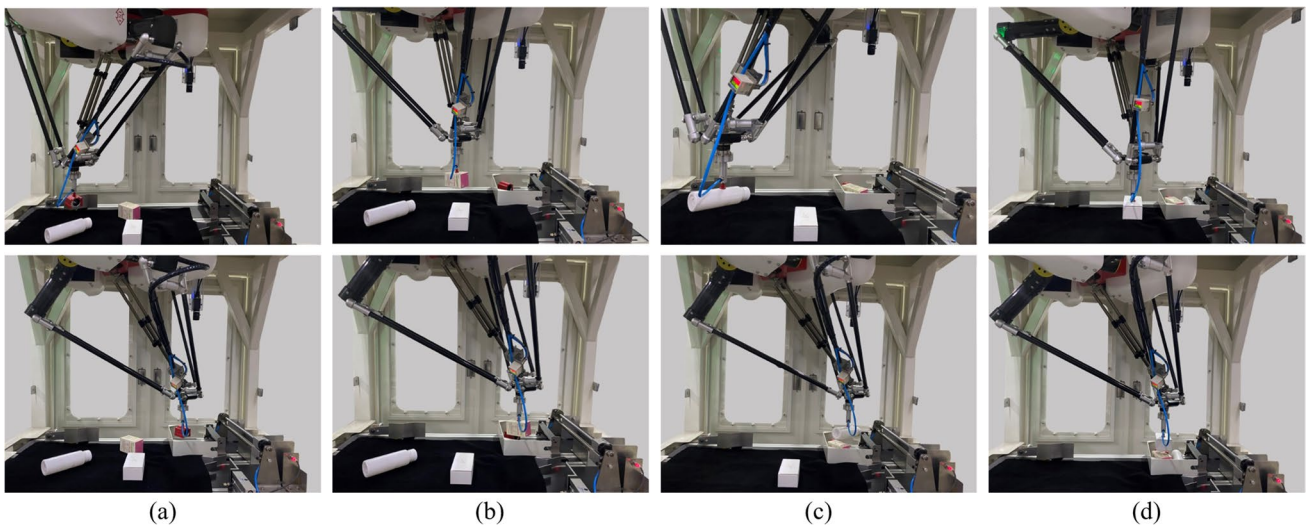
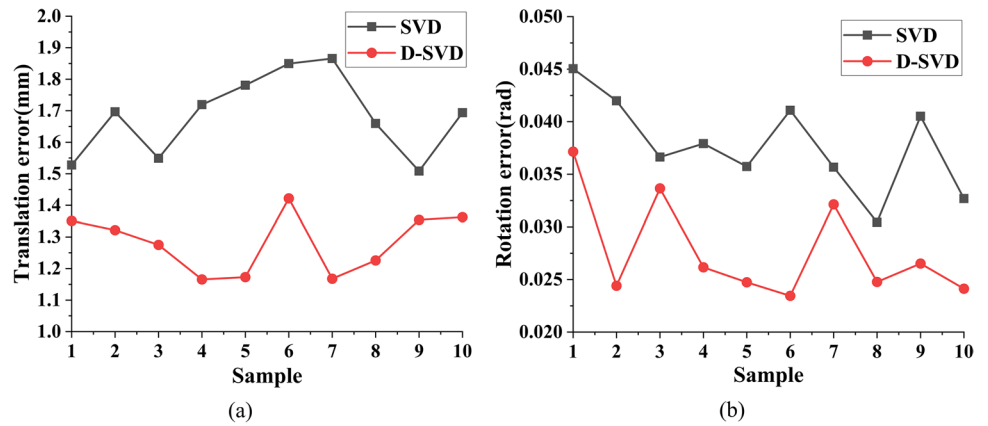


Fig. 20 Robot pickup operation. **a** Part. **b** Box1. **c** Bottle. **d** Box2

As shown in Fig. 19a, the calibration error of the 10 sample groups using the SVD method is large, ranging from 1.52 to 1.86 mm. The D-SVD method reduces the average translation error by 0.403 mm. As shown in Fig. 19b, the rotational error under the SVD method ranges from 0.035 to 0.045 rad, whereas the D-SVD reduces the average rotational error by 0.025 rad. The experimental analysis shows that D-SVD has a certain level of reliability.

4.4 Robot pickup results and evaluation

The test computer calculates the 3D pickup coordinate points, which are input into the robot demonstrator. For 50 groups of pickup experiments, the objects to be picked up are placed at various positions. The optimal pickup points are determined in the order of “part-box 1-bottle-box 2.” To prevent any contact with other objects during the pickup process, the robot performs a gate trajectory setup. The pickup results are shown in Fig. 20.

As shown in Fig. 20, the sequential pickup and placement process successfully positions all four objects into the designated area. The pickup errors mainly arise from the registration errors in the point cloud, calibration errors of the robot, and the inherent accuracy limitations of the robot itself, all

of which require thorough evaluation. To obtain accurate true coordinates, the end-effector is initially directed to the center of the target and the pickup point location. The robot demonstrates a positioning accuracy of approximately 0.05 mm. Subsequently, the calculated pickup coordinates are transmitted to the demonstrator using remote IP connectivity. The demonstrator controls the robot’s end-effector to realize the pickup operation. The objects are adjusted and placed in designated areas, and the effective positioning errors are analyzed through 50 groups of pickup experiments. Comprehensive error results are performed, including the positioning effective errors in the x -, y -, and z -axes and angle effective errors, as shown in Fig. 21.

Table 4 Time evaluation results

Target	Pickup rate (%)	Pick-up time (s)	Total inference time (s)
Part	86%	2.31	9.375
Box1	96%	2.23	10.011
Bottle	88%	2.42	9.752
Box2	98%	2.19	8.916

Fig. 21 Pickup positioning effective error results. **a** X-axis. **b** Y-axis. **c** Z-axis. **d** Angle

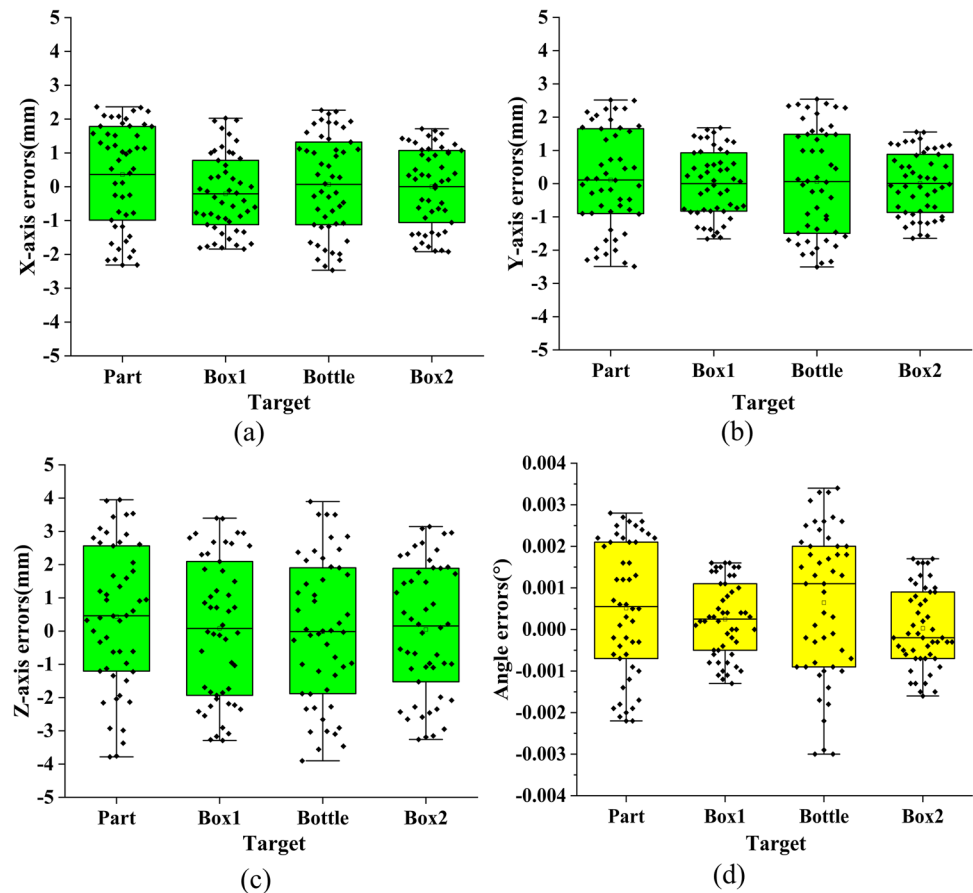
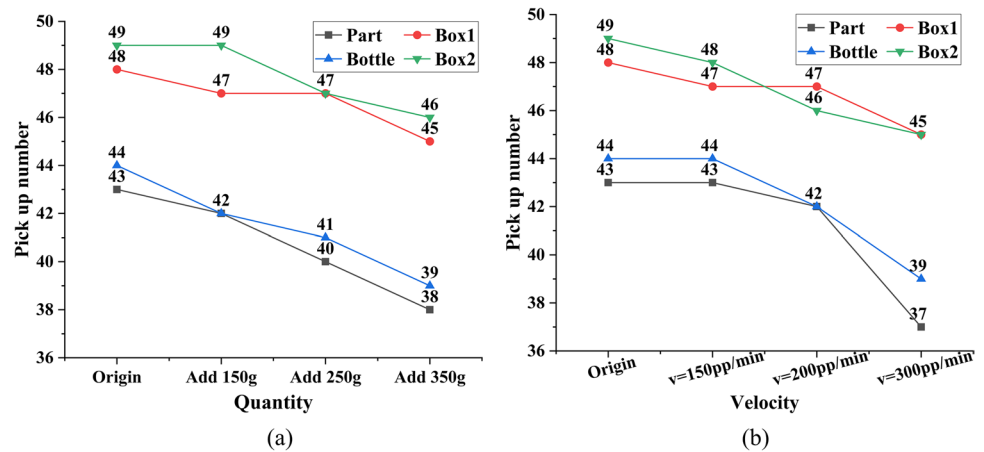


Fig. 22 Pickup evaluation result statistics. **a** Quantity. **b** Velocity

As shown in Fig. 21 a–c, the effective error statistics reveal that the positioning errors for part and bottle are relatively larger. This is mainly due to the incomplete mapping of their geometries in the point cloud, resulting in greater deviation between the calculated and actual pickup coordinates. The errors in the x - and y -axes for all four objects range between -2.32 and 2.48 mm, while the errors in the z -axis are larger, with a base error ranging between -3.79 and 3.95 mm. The positioning error is measured at 4.67 mm, while the robot angle error is $[-0.0028^\circ, 0.0034^\circ]$.

As shown in Table 4, box1 and box2 have the highest pickup rate, with a successful pickup rate of more than 96%. The part and bottle have a poor pickup effect due to the unevenness of the upper surface, and the successful pickup rate is around 86%. The total inference time contains the time required for all pre-processing operations of the point cloud. Pickup time is the time required for the end-effector to pick up from the camera area to the specified area. The basic pickup time is within 2.5 s, which is relatively short and in line with the fast pickup of parallel robots. The total inference time is around 10 s, real-time efficiency needs to be improved.

To assess the effectiveness of object pickup, the robot's running speed and the quality of the objects were adjusted. Object quality was altered by adding uniform weights, while maintaining the basic shape and size. The original weight of the objects ranged between 20 and 360 g, with additional weights of 150 g, 250 g, and 350 g. In order to test the pickup results under three different robot running speeds, the speeds were determined based on the frequency of the robot's movement within a fixed distance and time. The robot speeds were set to be $v = 140, 160,$ and 180 pp/min, with the distance from the objects' area to the designated area set at 620 mm. A total of 50 sets of pickup samples were established, and the pickup results are shown in Fig. 22.

As shown in Fig. 22, the success rate of picking up box 1 and box 2 remains relatively stable as the quality increases, while the pickup rates for bottle and part decline. At a

running speed of 150 pp/min, the number of successful pickups remained constant, with a significant decrease observed only when the running speed was increased to 300 pp/min. This outcome validates the reliability of the point cloud-based 3D pickup method.

5 Conclusion

In this paper, the 3D pickup estimation method for parallel robots based on point cloud simplification and registration is carried out. By improving traditional point cloud processing methods, a reliable 3D pickup estimation can be achieved. Point cloud pickup applied to parallel robots remains an area with limited research, and this study makes a preliminary attempt within industrial environment. Several methods have been explored in this study, including the FF-WLOP point cloud simplification method, the optimal projection point strategy, and the dynamic weighted hand-eye calibration. These methods prove to be reliable for the pickup operations of industrial parallel robots. The experimental results show an average point cloud registration error is 0.38 mm, an average hand-eye calibration error is 1.281 mm, and a true pickup above 92%, meeting the basic requirements of task.

Currently, due to the complexity of industrial environments, multiple filtering and simplification lead to low accuracy of target positioning. Real-time, end-to-end fast pickup operations have not been fully realized yet, although the prediction of multi-target depth information and complete 3D pickup can be achieved. The next research step in research will involve simplifying the pickup operation process to enhance inference speed and positioning accuracy. In future work, the robots will be deployed in industrial production environments equipped with high-speed conveyor belts to realize real-time pickup classification of high-speed dynamic objects. A gripper end-effector the advanced algorithms (e.g., GRCNN, PointNet) will be utilized to pick up irregular objects with non-uniform density.

Author contribution The manuscript was written by Yuting Zhang, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors wish to express their gratitude to the National Ministry of Science and Technology Innovation Method Special (2020IM020700).

Declarations

Competing interests The authors declare no competing interests.

References

- Tian H, Song K, Li S, Ma S, Xu J, Yan Y (2023) Data-driven robotic visual grasping detection for unknown objects: a7 problem-oriented review. *Expert Syst Appl* 211:118624. <https://doi.org/10.1016/j.eswa.2022.118624>
- Li X, Zhang X, Zhou X, Chen I-M (2023) UPG: 3D vision-based prediction framework for robotic grasping in multi-object scenes. *Knowl-Based Syst* 270:110491. <https://doi.org/10.1016/j.knosys.2023.110491>
- Diao S, Yang Y, Cui G, Chen Y (2023) Complex parts machining path planning through cross-source point cloud processing. *Comput Commun* 203:48–59. <https://doi.org/10.1016/j.comcom.2023.02.024>
- Kelaiaia R (2017) Improving the pose accuracy of the delta robot in machining operations. *Int J Adv Manuf Technol* 91:2205–2215. <https://doi.org/10.1007/s00170-016-9955-8>
- Liu C, Cao GH, Qu, YY et al (2020) An improved PSO algorithm for time-optimal trajectory planning of delta robot in intelligent packaging. *Int J Adv Manuf Technol* 107:1091–1099. <https://doi.org/10.1007/s00170-019-04421-7>
- Barnfather JD, Abram T (2018) Efficient compensation of dimensional errors in robotic machining using imperfect point cloud part inspection data. *Measurement* 117:176–185. <https://doi.org/10.1016/j.measurement.2017.12.021>
- Jiang H, Zhang T, Wachs JP, Duerstock BS (2016) Enhanced control of a wheelchair-mounted robotic manipulator using 3-D vision and multimodal interaction. *Comput Vis Image Underst* 149:21–31. <https://doi.org/10.1016/j.cviu.2016.03.015>
- Qiu S, Kermani MR (2022) A new approach for grasp quality calculation using continuous boundary formulation of grasp wrench space. *Mech Mach Theory* 168:104524. <https://doi.org/10.1016/j.mechmachtheory.2021.104524>
- Jerbić B, Šuligoj F, Švaco M, Šekoranja B (2015) Robot assisted 3D point cloud object registration. *Procedia Eng* 100:847–852. <https://doi.org/10.1016/j.proeng.2015.01.440>
- Zaki AMA, Carnevale M, Giberti H, Schlette C (2023) Digital model reconstruction through 3D stereo depth camera: a faster method exploiting robot poses. *Procedia Comput Sci* 217:1542–1549. <https://doi.org/10.1016/j.procs.2022.12.354>
- Li Z, Xu B, Wu D, Zhao K, Chen S, Lu M, Cong J (2023) A YOLO-GGCNN based grasping framework for mobile robots in unknown environments. *Expert Syst Appl* 225:119993. <https://doi.org/10.1016/j.eswa.2023.119993>
- Kim P, Chen J, Cho YK (2018) SLAM-driven robotic mapping and registration of 3D point clouds. *Autom Constr* 89:38–48. <https://doi.org/10.1016/j.autcon.2018.01.009>
- Han X, Jin JS, Wang M, Jiang W, Gao L, Xiao L (2017) A review of algorithms for filtering the 3D point cloud. *Signal Process: Image Commun* 57:103–112. <https://doi.org/10.1016/j.image.2017.05.009>
- Sun G, Chu C, Mei J, Li W, Su Z (2022) Structure-aware denoising for real-world noisy point clouds with complex structures. *Comput Aided Des* 149:103275. <https://doi.org/10.1016/j.cad.2022.103275>
- Liu Y, Tang Q, Tian X, Yang S (2023) A novel offline programming approach of robot welding for multi-pipe intersection structures based on NSGA-II and measured 3D point-clouds. *Robot Comput-Integr Manuf* 83:102549. <https://doi.org/10.1016/j.rcim.2023.102549>
- Zhuang C, Li S, Ding H (2023) Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking. *Robot Comput-Integr Manuf* 82:102541. <https://doi.org/10.1016/j.rcim.2023.102541>
- Faria DR, Trindade P, Lobo J, Dias J (2014) Knowledge-based reasoning from human grasp demonstrations for robot grasp synthesis. *Robot Auton Syst* 62(6):794–817. <https://doi.org/10.1016/j.robot.2014.02.003>
- Qi CR, Su H, Mo K, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. <https://doi.org/10.48550/arXiv.1612.00593>
- Qi CR, Li Y, Hao S, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space. <https://doi.org/10.48550/arXiv.1706.02413>
- Liang H, Ma X, Li S, Grner M, Tang S, Fang B, Sun F, Zhang J (2019) PointNetGPD: detecting grasp configurations from point sets. <https://doi.org/10.48550/arXiv.1809.06267>
- Ni P, Zhang W, Zhu X, Cao Q (2020) PointNet++ grasping: learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, pp 3619–3625. <https://doi.org/10.1109/ICRA40945.2020.9196740>
- Tian H, Song K, Xu J, Ma S, Yan Y (2023) Antipodal-points-aware dual-decoding network for robotic visual grasp detection oriented to multi-object clutter scenes. *Expert Syst Appl* 230:120545. <https://doi.org/10.1016/j.eswa.2023.120545>
- Park J, Jun MBG, Yun H (2022) Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (CNN). *J Manuf Syst* 63:539–549. <https://doi.org/10.1016/j.jmsy.2022.05.011>
- Eguíluz AG, Rañó I (2022) Heuristic grasping of convex objects using 3D imaging and tactile sensing in uncalibrated grasping scenarios. *Expert Syst Appl* 208:118114. <https://doi.org/10.1016/j.eswa.2022.118114>
- Zhang T, Zhang C, Hu T (2022) A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios. *Robot Comput Integr Manuf* 76:102329. <https://doi.org/10.1016/j.rcim.2022.102329>
- Zhang T, Zhang C, Shuaiji THu (2023) Robot suction region prediction method from knowledge to learning in disordered manufacturing scenarios. *Eng Appl Artif Intell* 120:105928. <https://doi.org/10.1016/j.engappai.2023.105928>
- Quanquan S, Hu J, Wang W, Yi F, Liu W, Qi J, Ma J (2019) Suction grasp region prediction using self-supervised learning for object picking in dense clutter. In: IEEE 5th International Conference on Mechatronics System and Robots (ICMSR), Singapore, 2019, pp 7–12. <https://doi.org/10.1109/ICMSR.2019.8835468>
- Li J, Guo S, Meng X, Lai Z, Han S (2022) DPG-Net: densely progressive-growing network for point cloud completion. *Neurocomputing* 491:1–13. <https://doi.org/10.1016/j.neucom.2022.03.060>
- Zhou Y, Chen R, Zhao Y, Ai X, Zhou G (2021) Point cloud denoising using non-local collaborative projections. *Pattern Recogn* 120:108128. <https://doi.org/10.1016/j.patcog.2021.108128>
- Bellakehal S, Andreff N, Mezouar Y, Tadjine M (2011) Vision/force control of parallel robots. *Mech Mach Theory* 46(10):1376–1395. <https://doi.org/10.1016/j.mechmachtheory.2011.05.010>

31. Wei P, Yu X, Di Z, Dai X, Wang B, Zeng Y (2022) Design of robot automatic navigation under computer intelligent algorithm and machine vision. *J Ind Inf Integr* 28:100366. <https://doi.org/10.1016/j.jii.2022.100366>
32. Yang H, Chen L, Ma Z, Chen M, Zhong Y, Deng F, Li M (2021) Computer vision-based high-quality tea automatic plucking robot using delta parallel manipulator. *Comput Electron Agric* 181:105946. <https://doi.org/10.1016/j.compag.2020.105946>
33. Korayem MH, Yousefzadeh M, Kian S (2018) Precise end-effector pose estimation in spatial cable-driven parallel robots with elastic cables using a data fusion method. *Measurement* 130:177–190. <https://doi.org/10.1016/j.measurement.2018.08.009>
34. Goodwin J, Saldaña C (2023) Vision-based localization for cooperative robot-CNC hybrid manufacturing. *Int J Adv Manuf Technol* 126:241–258. <https://doi.org/10.1007/s00170-023-11009-9>
35. Xue K, Wang Z, Shen J, Shengsun Hu, Zhen Y, Liu J, Dingyong Wu, Yang H (2021) Robotic seam tracking system based on vision sensing and human-machine interaction for multi-pass MAG welding. *J Manuf Process* 63:48–59. <https://doi.org/10.1016/j.jmapro.2020.02.026>
36. Santhakumar K, Kasaei H (2022) Lifelong 3D object recognition and grasp synthesis using dual memory recurrent self-organization networks. *Neural Netw* 150:167–180. <https://doi.org/10.1016/j.neunet.2022.02.027>
37. Moosmann M, Spenrath F, Kleeberger K, Usman Khalid M, Mönig M, Rosport J, Bormann R (2020) Increasing the robustness of random bin picking by avoiding grasps of entangled workpieces. *Procedia CIRP* 93:1212–1217. <https://doi.org/10.1016/j.procir.2020.03.082>
38. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D ShapeNets: a deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, pp 1912–1920. <https://doi.org/10.1109/CVPR.2015.7298801>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.