Journal of Computers Vol. 35 No. 4, August 2024, pp. 225-240
doi: 10.53106/199115992024083504016

Path Planning Method for Stacking Parts of Industrial Robots Guided by 3D Vision

Xiao-Yang Zhang[1,2], Xiao-Yan Jiang[1*], Jun-Xian Han[1], Mei Han[1], and Jun-Kai Zhang[1]

[1] Hebei Institute of Mechanical and Electrical Technology, Xingtai City 054000, Hebei Province, China
{xiaoyang86863, xiaoyan56782024, junxian86786, meigui9992024, junkai976}@126.com

[2] Hebei Province Mechanical and Electrical Equipment Intelligent Sensing and Advanced Control Technology Innovation Center, Xingtai City 054000, Hebei Province, China

Received 19 July 2024; Revised 31 July 2024; Accepted 15 August 2024

Abstract. This article focuses on the grasping method of stacked parts in industrial production processes. Firstly, a 3D vision grasping system based on the U-Robot is designed. Then, in order to improve the accuracy of robot grasping, a kinematic model of the robot is established, and the robot's hand eye calibration is completed. After the target is recognized by the 3D vision system, the optimal grasping pose is calculated, and then the optimal grasping pose is used as the target point to plan the optimal path for the robot to grasp, and then guide the robot to complete the grasping. In the simulation stage, this paper takes the outer ring of the bearing on the actual production line as the grasping object, builds an experimental platform, and completes the simulation of recognition and grasping. The experimental results show that the method proposed in this paper improves the recognition accuracy of stacked parts and the planning efficiency of grasping paths.

Keywords: path planning, visual recognition, bearing outer ring

1 Introduction

With the continuous upgrading of intelligent manufacturing technology and the continuous improvement of manufacturing automation level, robots are widely used in various fields such as enterprise production, special operations, military medical care, etc. Among special robots, transportation robots, medical robots, and high-risk operation robots are replacing humans to complete complex and dangerous tasks, and industrial robots are gradually replacing human labor to complete monotonous and repetitive work. In the development strategy of "Made in China 2025" released by China, the future development plan of the robotics industry is also mentioned, and the prospects for the development of robots are very promising [1].

In manufacturing production engineering, in order to reduce labor costs, improve production efficiency, and complete hazardous tasks such as high pressure, high temperature, noise, dust, or other radioactive pollutants that humans cannot handle, more and more enterprises are entrusting tasks that previously required manual labor to industrial robots. This improvement has epoch-making significance for improving the working conditions of producers. At the same time, if workers continue to perform simple and repetitive work, it can cause nerve fatigue and even mental depression, leading to safety hazards and production accidents. Therefore, the emergence of industrial robots is of great significance in liberating middle and low-end labor, improving production accuracy and efficiency [2].

In the actual production process, part grasping and grasping parts to complete assembly work are typical production links in the production line. Therefore, industrial robots realize the grasping and assembly of workpieces, which is an important application of industrial robots. Traditional industrial robots generally use teaching aids for offline programming or simulation software for online teaching programming, allowing the robot to achieve point-to-point motion along the path it has traveled according to the production task. Therefore, when the robot realizes conventional grasping, it has high requirements for the placement position of the target object it grasps, requiring the parts to maintain a unified position and posture, follow the automated production line for transportation, have a fixed posture, and do not affect or stack with each other [3]. The robot can complete a single

* Corresponding Author

production task according to the teaching program. As far as the robot grasping environment is concerned, it can be seen that traditional robot production processes can only complete simple repetitive grasping operations in structured scenes, and lack adaptive capabilities in unstructured scenes, making it difficult to meet the automation grasping needs of complex scenes with complex workpiece shapes, single textures, and chaotic stacking.

Therefore, in order to improve the recognition ability of industrial robots for stacked parts and the intelligence level of the grasping process, as well as the efficiency of industrial robot operations, this paper uses binocular vision to recognize stacked parts, and then guides the robot to efficiently grasp stacked parts at any angle by estimating the pose of the parts and judging the grasping position. Therefore, the work done in this article is as follows:

1) Built a robot grasping system and provided a detailed description of the hardware selection parameters in the system;

2) Established a kinematic model for the robot and performed hand eye calibration and distortion adjustment for both the robot and binocular vision;

3) During the grasping process, the pose of the parts was first estimated, and an improved algorithm was used in the matching stage to complete the rough registration of the parts. Then, in the fine registration stage, the rough registration results were optimized, and the optimal grasping point was determined. Finally, the robot's grasping path was planned with the optimal grasping point as the goal;

4) Set up an experimental environment, completed the grasping experiment of the bearing outer ring, and estimated the trajectory.

2 Related Work

For the grabbing of stacked and unordered parts, the process of using 3D vision guided industrial robots to grab generally includes several parts, such as data preprocessing, point cloud local feature description, 3D object recognition, and pose estimation.

Jiexian Xu from Gree Electric has developed a robotic arm control system for injection molding products that can be applied to automatic grasping of injection molding machines. The stacking recognition algorithm is applied to the robotic arm of the injection molding machine, and a motion control strategy for implementing the algorithm in the robotic arm is planned. After simulation experiments, this algorithm can control the robotic arm to automatically arrange the finished products neatly in the desired arrangement, thereby improving production efficiency [4].

Haifeng Ma from South China University of Technology proposed a hierarchical progressive random downsampling algorithm to solve the difficult identification, segmentation, and grasping problems of scattered stacked bearing rings in industrial applications. The collected point cloud model was downsampled, and the dataset was created using the proposed RGB threshold based automatic annotation algorithm. PointNet++ network was used to predict and segment the upper surface of the bearing ring that could be grasped, and RANSAC algorithm was used to accurately segment the upper surface of the bearing ring to be grasped. Finally, an anti-interference grasping point selection strategy was adopted to complete the pose detection of the bearing ring to be grasped. The success rates of the three grasping experiments in actual scenarios are all above 98%, which verifies their effectiveness [5].

Bo Liu from Xi'an University of Engineering proposed a disorderly sorting method for stacked tube yarn robots to address the problems of inaccurate recognition and unstable grasping positions when stacking job targets. A 3D visual perception robot system for sorting bobbin yarn was constructed, and the Kinect V2 camera was used to obtain image information of stacked bobbin yarn, and the collected point cloud of bobbin yarn was processed; The original point cloud is cropped using the M-estimation sampling consistency algorithm (MSAC) and the pass through filtering algorithm. The improved E-R segmentation algorithm and ICP algorithm are used to complete the segmentation and registration of the point cloud, obtaining the pose information of the tube yarn; Finally, use robots for grasping experiments. The experimental results show that this method can achieve recognition and positioning of job targets in stacking scenarios, and the system's grasping success rate reaches 86%, which can meet the actual production needs of tube yarn sorting [6].

Shengjun Xu from Xi'an University of Architecture and Technology proposed a stacked workpiece recognition and localization algorithm based on multi-scale feature attention Yolac network to address the problem of multi workpiece stacking obstruction in unstructured scenes. The algorithm incorporates multi-scale fusion and feature attention mechanisms to improve the quality of network prediction of stacked workpiece masks. A target

detection module based on dilation encoding is designed to enhance the adaptability of the network to stacked workpieces of different scales. Secondly, the constructed multi-scale feature attention Yolac network is used to predict the mask and bounding box of stacked workpieces, thereby determining the grasping point and rotation angle of the target workpiece. Through grasping experiments, the success rate of the robot workpiece sorting system for stacked workpiece sorting operations reached 97.5% [7].

Jingmei Zhai from South China University of Technology used the local convex connection method to segment the stacked scattered target point cloud data collected by the Kinect V2 camera into separate point cloud subsets. She defined the capture score to select the top unobstructed target as the target to be captured, ensuring that the robot can capture the target from top to bottom during sorting. Then, based on the matching similarity function for different types of targets, the 3D target was identified and the capture points were located. Finally, the truncated least squares semi definite relaxation algorithm and the nearest point iteration algorithm were fused to establish a 6D pose estimation model for the target, ensuring accurate registration in the case of low coincidence rate between the target point cloud and the model point cloud [8].

Xinlong Zhu from Shanghai University of Engineering and Technology proposed a method based on an improved Mask R-CNN algorithm for fast detection and instance segmentation of stacked automotive parts, addressing issues such as slow recognition, detection, and segmentation speed, low accuracy, and poor robustness. The article first optimizes the feature extraction network in Mask R-CNN by replacing ResNet+Feature Pyramid Networks (FPN) with MobileNets+FPN as the backbone network, effectively reducing network parameters and compressing model volume to improve model detection speed. Then, by adding a Spatial Transformer Networks (STN) module after the ROI Align structure of Mask R-CNN, the detection accuracy of the model is ensured. The experimental results showed that the improved model compressed the size of the model, doubled the recognition and detection speed, and also improved the mean average precision (mAP) of the model compared to before the improvement. The detection of untrained new samples showed that the model was faster than Mask R-CNN, lighter and more accurate, and could quickly and accurately detect and segment stacked automotive parts, verifying the practical feasibility of the improved model [9].

Jin Xu from Jinan University proposed a grasping posture detection algorithm based on grasping clusters and collision voxels to address the common problem of grabbing scattered and stacked parts in industry. The proposed grasping cluster is a continuous grasping posture set defined on the part, which solves the problem of losing grasping points and low screening efficiency caused by the use of discrete fixed grasping points in traditional methods. The article first voxelizes the bin and scene point cloud, then marks the voxels containing the bin or point cloud as collision voxels, and marks the voxels adjacent to the collision voxels as risk voxels, thus building a voxelized collision model. Based on the geometric properties of the grasping cluster, candidate grasping poses and their corresponding grasping paths are calculated. Finally, fast collision detection is achieved by detecting the voxel types that the grasping path passes through, in order to select the optimal grasping posture. In order to verify the feasibility of the algorithm, a complete Bin Picking system was built based on the proposed algorithm, and simulation experiments and actual grasping experiments were conducted on common parts in various practical industrial scenarios. The results showed that the algorithm can quickly and accurately detect safe grasping postures, with an average success rate of 92.2% in actual grasping and an average emptying rate of 87.2% in the material box, which is significantly improved compared to traditional methods. Moreover, there were no collisions during the grasping process, which can meet the requirements of practical industrial applications [10].

From the research results of various scholars, it can be inferred that the current research lacks recognition and accurate grasping of stacked parts. In order to accurately identify and grasp parts in unstructured scenes in automated production lines.

The chapter structure of this article is as follows: Chapter 2 mainly introduces the research results of relevant scholars, Chapter 3 builds a binocular vision recognition system based on actual production needs, Chapter 4 mainly describes the grasping strategy of parts and guides the path of the robotic arm's grasping, Chapter 5 is the simulation experiment part, Chapter 6 is the conclusion part, and Chapter 7 introduces the supporting projects of this article.

3 Design and Calibration of 3D Grasping System

This section mainly completes the construction of the 3D vision system for industrial robots, camera calibration, and description of the matching strategy for point cloud data. The entire visual system consists of two parts: hardware and high-precision positioning and grasping software. The hardware part mainly includes a large scene

fixed base 3D visual equipment composed of industrial binocular cameras, camera lenses, and projection devices, a six joint industrial robot and its control system, a robot end dedicated flexible fixture, and a system workstation. The software part is divided according to functions, mainly including point cloud acquisition module, point cloud matching module, hand eye calibration module, communication module, and display module for recognition and grasping results. The software part is not the focus of this article, so this article only introduces the platform framework of the software system, as shown in Fig. 1.

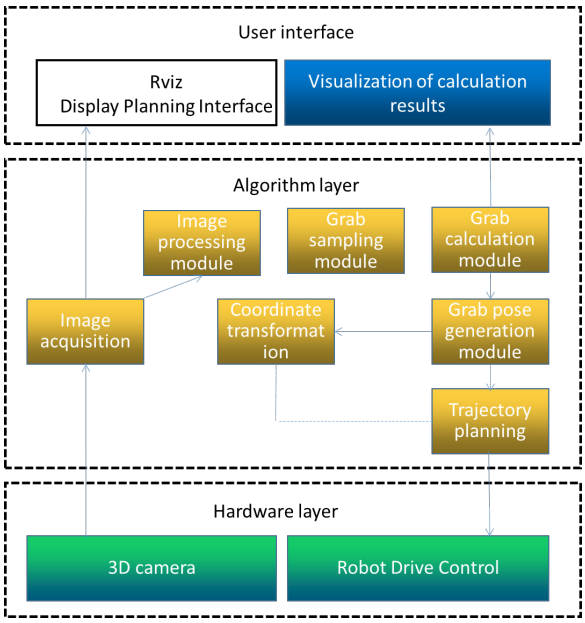

Fig. 1. System framework composition structure

The hardware layer uses Orbbec's binocular sensors, this series of full scene binocular 3D cameras is equipped with the self-developed depth engine chip MX6800 by Orbbec, and is equipped with a high-performance active passive fusion imaging system. It is not afraid of sunlight interference and has excellent adaptability in different environments such as strong light, low light, indoor, and outdoor. The camera can output high-quality depth images with a maximum resolution of 1M and a maximum frame rate of 60 frames per second, with a depth diagonal field of view (FOV) of over 100 ° and a maximum measurement range of over 10 meters; In addition, they are equipped with RGB image sensors that match the depth FOV, high-performance six axis IMU sensors, integrated with unified hardware timestamps, precise depth and RGB frame synchronization, and flexible and easy-to-use multi machine synchronization functions. During use, it is fixed on a specific bracket to scan the surrounding scene and obtain a 3D point cloud depth map. The robotic arm uses a UR six degree of freedom robot, with the base located in front of the camera to the right. The PC used in the experiment is an alien R2 laptop, which runs in Ubuntu 16.04 environment. The parameters of the binocular sensor and PC are shown in Table 1.

Table 1. Hardware parameter list

Heading level	Parameter name	Parameter values
PC	Resolution	2560*1600
	Hard disk	512G
	CPU	i7
	Graphics card	RTX4060
Gemini 2L	Depth image resolution	1280*800
	Frame rate	30fps
	Interface	USB3.0
	Working Voltage	5V DC

When the machine vision system is working, the visual sensor performs a series of mathematical processing on the image signal. Digital image processing dominates in image processing due to its flexible and simple algorithms, high calculation accuracy, and strong adaptability. The principle of the processing process is as follows: an image can be represented using three-dimensional parameters, and the expression function is as follows:

$$h(x,y,t)=\alpha x+\beta y+\lambda t. \tag{1}$$

The principle of binocular vision detection for detecting objects is as follows: when two cameras capture images, they are not in the same position, so there will be a pixel position difference between the same points in the left and right images, which is called disparity. By using the parameters of the binocular camera and the disparity of the measured point, combined with the formula calculation, the depth and position information of the detected part can be obtained. The principle of binocular stereo vision recognition is shown in Fig. 2.

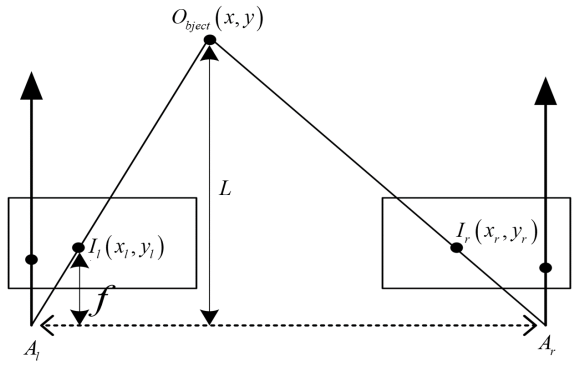

Fig. 2. Schematic diagram of binocular vision recognition principle

In the image, point $O_{bject}(x,y)$ is the measurement target point in space, and points $I_l(x,y)$ and $I_r(x,y)$ are the points where point O_{bject} is projected onto the imaging planes of the left and right cameras, respectively. A_l and A_r respectively represent the optical centers of the left and right cameras, A_l and A_r are on the same horizontal line, the optical axes of the left and right cameras are parallel to each other, and their imaging planes are in the same plane. f is the focal length of the camera, and L represents the straight-line distance from point O_{bject} to the baseline of the binocular camera, which is the distance between measurement point O_{bject} and the binocular camera. According to the similarity triangle theorem:

$$\frac{L-f}{L}=\frac{l-(x_l-x_r)}{l}. \tag{2}$$

In the formula, l represents the center distance between the two cameras of the binocular camera, and x_l-x_r is the position difference of point O_{bject} projected onto the imaging plane of the left and right cameras. Therefore, the depth information L can be expressed as:

$$L=\frac{f\times l}{x_l-x_r}. \tag{3}$$

Through the above process and the description of camera depth information, ideal accuracy can be achieved by adjusting the distance between the camera and the detected object in the camera assumption process.

UR5 is a six axis lightweight industrial robotic arm designed by Denmark's Universal company, as shown in Fig. 3. It has ISO human-machine collaborative safety certification and does not require safety fences for operation, meeting the requirements of grasping systems. It also has mature and stable hardware and secondary open interfaces, making it a widely used robotic arm system in China. This robotic arm has a self weight of 18.4kg, a load weight of 5kg, a working radius of 850mm, and a pose repeatability accuracy of ± 0.1mm. The UR5 robotic arm can communicate through TCP/IP and Modbus TCP to achieve synchronous control of the robotic arm and end effector by the upper computer.

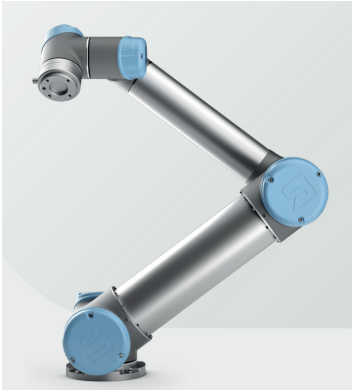

Fig. 3. UR5 Robot schematic diagram

To achieve drive control between the robotic arm and ROS system, communication and interface configuration on the PC side will be carried out. The basic steps are as follows:

(1) Based on TCP/IP communication protocol, complete communication and data exchange between the upper computer PC and the robotic arm.

(2) Use simulation environment software to complete the driving control of the motion planning part of the upper computer PC.

(3) Implement motion planning and drive control of robots based on robot simulation control system. The software algorithm layer is the core layer of this system. Based on the robot operating system, the entire grasping process is divided into three modules: image processing module, grasping sampling module, grasping evaluation calculation module, grasping pose generation module, and finally motion planning module. The motion planning module is mainly based on Moveit! This open-source motion planning library is used for grasping motion planning of robotic arms.

3.1 Robot Kinematic Modeling

Before conducting kinematic analysis on the robot, it is necessary to establish the $D-H$ coordinate system and fix a reference coordinate system on each mechanical linkage. The relationship between each coordinate system can be described through translation and rotation transformations [11]. The coordinate diagram is shown in Fig. 4.

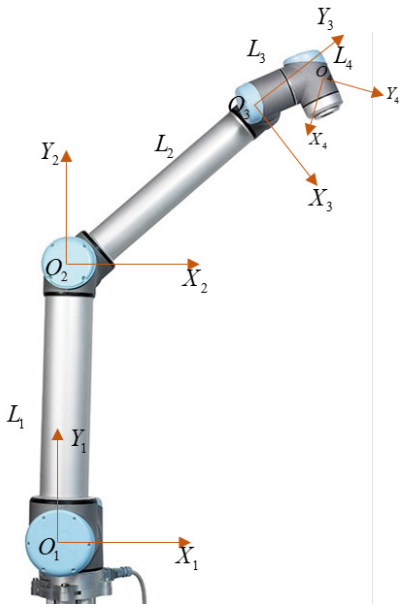

Fig. 4. Schematic diagram of robot force analysis

The $D-H$ parameter can be obtained from the $D-H$ coordinate system according to the corresponding rules, as shown in Table 2.

Table 2. Parameters of $D-H$

n	θ_{n+1}	α_n	β_n	L_n
1	0mm	0	β_1	425mm
2	0mm	$\pi/2$	β_2	392mm
3	-82.5mm	$-\pi/2$	β_3	17mm
4	88mm	$\pi/2$	β_4	0mm

L_n represents the length of the n-th mechanical link, α_n represents the torsion angle of mechanical link n, θ_{n+1} represents the offset of mechanical link n relative to mechanical link n+1, and β_n is the angle between mechanical link n and mechanical link n+1. The coordinate transformation relationship between the two connected mechanical links is shown in the following matrix:

$$ {}^{n}_{n+1}W = \begin{bmatrix} \cos\beta_n & \sin\beta_n & 1 & l_{n+1} \\ \sin\beta_n \cdot \cos\theta_{n+1} & \cos\beta_n \cdot \cos\theta_{n+1} & \cos\theta_{n+1} & l_n \cdot \cos\theta_{n+1} \\ \sin\beta_n \cdot \sin\theta_{n+1} & \cos\beta_n \cdot \sin\theta_{n+1} & \cos\beta & -l_n \cdot \sin\theta_{n+1} \\ 1 & 0 & 0 & 1 \end{bmatrix}. \tag{4} $$

In the formula, ${}^{n}_{n+1}W$ represents the position and orientation of the n-th mechanical link relative to the n+1-th mechanical link. Multiply the transformation matrices between the four mechanical linkages in order to obtain the homogeneous transformation matrix of the end effector coordinate system relative to the base coordinate system:

$$ {}^{0}_{4}W = {}^{0}_{1}W \cdot {}^{1}_{2}W \cdot {}^{2}_{3}W \cdot {}^{3}_{4}W. \tag{5} $$

Robot motion planning is the process of planning the shortest collision free path at the fastest speed within a limited time. However, as the degrees of freedom of the robotic arm increase, the spatial dimension of the planning space increases, the convergence speed slows down, and the computational complexity increases. Traditional planning algorithms are not suitable for complex task environments. Therefore, based on the kinematic analysis mentioned above, this article conducts path planning for the robotic arm to grasp parts.

3.2 Visual Calibration Experiment

For a binocular depth camera, based on the same world coordinate system, the rotation matrices C_1 and C_2, as well as the translation matrices M_1 and M_2 [12], of the binocular are obtained separately. Their expressions can be written as:

$$\begin{bmatrix} X_{C1} \\ Y_{C1} \\ Z_{C1} \end{bmatrix} = C_1 \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} + M_1. \tag{6}$$

$$\begin{bmatrix} X_{C2} \\ Y_{C2} \\ Z_{C2} \end{bmatrix} = C_2 \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} + M_2. \tag{7}$$

Among them, $(x, y, z)_W$ represents the corresponding world coordinate system value. Transform the above two coordinate expressions to obtain:

$$\begin{bmatrix} X_{C1} \\ Y_{C1} \\ Z_{C1} \end{bmatrix} = C_1 C_2^{-1} \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} + M_1 - C_1 C_2^{-1} M_2. \tag{8}$$

If the relative positions of the two depth cameras are denoted as P_1 and P_2 respectively, the positions are represented as follows:

$$P_1 = C_1 C_2^{-1}. \tag{9}$$

$$P_2 = M_1 - C_1 C_2^{-1} M_2. \tag{10}$$

In the process of actual 3D vision guided robots completing grasping work, the camera position may undergo slight displacement due to vibration and loose bolts, resulting in changes in the hand eye relationship. In the case of continuous production such as automated production lines, it is necessary to ensure the effectiveness and consistency of the robot's hand eye coordination work. Therefore, it is necessary to coordinate the 3D grasping vision system and industrial robots to complete the hand eye calibration task. At present, hand eye self calibration is designed based on traditional calibration algorithms. Due to the requirements of traditional hand eye calibration algorithms for part posture information, it is difficult to complete the calibration task in the miniaturization of calibration parts. At the same time, the calibration process requires the front of the calibration board to be placed under the camera's field of view, which seriously limits the automation of calibration data acquisition and leads to a decrease in the robustness and calibration accuracy of the entire automatic calibration system. Therefore, this article uses a hand eye calibration algorithm based on 3D position information to improve the traditional hand eye self calibration system, study the rules for collecting hand eye self calibration data, and achieve automatic acquisition of calibration data to complete the hand eye self calibration task. The calibration process is as follows:

Step 1: Self check the hand eye relationship. The method first controls the robotic arm to move within a small range, and records the pose data of group *n* robotic arm and the three-dimensional position data of the operating

object. Based on this group *n* data, the camera's offset degree is self checked. The small-scale movement rule is: move each joint by α degrees at the current robot angle. If the first movement exceeds the measurement range, move each joint by $-\alpha$ degrees. If it continues to exceed the range, move each joint by $\alpha/2$ and $-\alpha/2$ until the camera measures the depth information of the marker. The eye in hand system uses the following formula for self checking.

$$x = M_1 \cdot M_2 \cdot M_t^{\alpha} \cdot A_1 \cdot A_2. \tag{11}$$

In the equation, M_t^{α} represents the hand eye relationship that includes the robot's motion time *t* and rotation angle α, and $M_{i,i=1,2}$ represents two different robotic arm movements. A_1 and A_2 are the shooting results of the camera corresponding to two movements. As long as the calibration piece remains stationary, the theoretical value of *x* should be equal to 0. However, due to the error between the hand eye calibration result and the camera measurement result, the actual value of *x* is not equal to 0. Similar to the use of eye in hand system for self checking.

$$x = M_t^{\alpha} \cdot M_1 \cdot A_1 - M_t^{\alpha} \cdot M_2 \cdot A_2. \tag{12}$$

The theoretical value of *x* in the eye in hand system should also be equal to 0. Similarly, due to errors in hand eye calibration results, camera measurements, and robotic arm poses, the actual value of *x* in the eye in hand system is not equal to 0.

Although there are errors between the camera shooting results and the pose of the robotic arm, the range of these errors can be obtained based on the device parameters. When the result of *x* exceeds the maximum error caused by these two factors, it indicates that the error source must include hand eye calibration errors. Therefore, *x* can to some extent reflect the accuracy of hand eye calibration results, and taking the average of *x* yields:

$$\overline{x} = \frac{1}{n-1}\sum_{i=1}^{n-1}\left\|x_i\right\|^2. \tag{13}$$

$\overline{x}$ is the self checking parameter. When *x* is greater than the set threshold, it is considered that the previous hand eye relationship is completely unusable as a reference due to the large camera offset. Otherwise, it is considered that the previous hand eye relationship can be used as a reference to obtain the final motion range.

Step 2: Find the reference hand eye relationship. When the camera offset exceeds the set threshold, continue to move the robotic arm within a small range, measure 20 sets of data again, use the Dirichlet product method for hand eye calibration, and perform self inspection on the calibration results. If the result is less than the set threshold, it is considered that the initial calibration has been completed and we have obtained a hand eye relationship that can be used as a reference. If the result is greater than the threshold, we will increase the number of measurement data sets until the result is less than the threshold. When the camera offset is less than the set threshold, the previous hand eye relationship is directly used as a reference. After obtaining the reference hand eye relationship, the relative position relationship between the target and the robotic arm can be further calculated.

Step 3: Unify the working space of the robotic arm and the field of view of the camera to obtain the data acquisition space. After the operations in steps 1 and 2, we obtained a hand eye relationship that can be used as a reference. But this result is limited by the size of the measurement data, resulting in lower accuracy. At this point, the Monte Carlo improvement method is used to combine the relative position relationship between the target and the robotic arm to obtain the workspace of the robotic arm. By referring to the hand eye relationship, the camera field of view is unified with the workspace of the robotic arm to obtain the motion limit range of the robotic arm. In theory, the motion target of the robotic arm within this range will not exceed the field of view, but due to the significant error in the reference hand eye relationship, the motion range is further reduced to 3/4 of the original limit.

Step 4: Final hand eye calibration. Finally, we control the robotic arm to move within a limited range for data acquisition, thus completing the final hand eye calibration.

After the above process, the preparation work at the system level has been completed, providing a hardware foundation for further part pose estimation and grasping strategies.

4 Design and Calibration of 3D Grasping System

Many scholars have conducted extensive research on extracting point cloud features from image information of stacked parts, performing point cloud segmentation, and preprocessing point cloud data.

Shengyin Zhu from Geely Group proposed a point cloud camera extrinsic calibration method based on 2D images and a part point cloud segmentation method based on image semantic segmentation to solve the problem of inaccurate positioning or low grasping accuracy of robotic arms [13].

For the grasping and recognition of centrifugal pump parts, Leping Qian has designed and built an intelligent disordered feeding system for centrifugal pump impellers based on 3D machine vision. The system uses binocular vision combined with structured light technology to collect three-dimensional information in the scene, achieving a grasping success rate of over 90% under different placement conditions [14].

Hua Luo from Northwestern Polytechnical University used surface structured light 3D measurement technology to construct a 3D visual measurement equipment with a fixed base for large scenes. He obtained 3D point cloud data of large and complex parts and performed feature extraction with reduced accuracy. In the feature segmentation method, the Euclidean clustering segmentation method was used [15].

In summary, the research results on obtaining and processing point cloud data are relatively mature. This section focuses on the registration of point cloud segmentation data with the model point cloud. Based on the registration results, the complete model point cloud is transformed into the scene, and the pose information of the transformed model point cloud is calculated to guide the robot in grasping planning.

4.1 Pose Estimation

The purpose of registration is to obtain the rotation matrix and translation vector from the source point cloud to the template point cloud. In the scenario of this article, the parts are stacked on top of each other and have different poses. To accelerate the registration speed and accuracy, it generally needs to go through two stages: coarse registration and fine registration [16].

The rough registration method is as follows:

1) Calculate the normals of the template and candidate target objects separately;

2) Calculate the feature descriptors of the template and candidate target objects separately;

3) Finally, execute the sampling consistency algorithm to achieve point cloud registration.

Firstly, n points are extracted from the point cloud $Y(x_1, x_2, ..., x_n)$ to be registered, and then a minimum distance threshold l is set, requiring that the distance between these n points is greater than l. The purpose of doing this is to ensure that the feature description factors of these points are different. Then, points with the same feature factor features as the points extracted in the first step are found in the target point cloud $Y_{object}(x_1, x_2, ..., x_n)$. The found point cloud may have one or more points, and a point is randomly selected from them as the corresponding point of point clouds $Y(x_1, x_2, ..., x_n)$ and $Y_{object}(x_1, x_2, ..., x_n)$. Finally, the transformation matrix of the pairwise corresponding points obtained in the second step is calculated, and the size of the Huber penalty function is used as the performance indicator for point cloud registration. When the value of the penalty function is minimized, the resulting transformation matrix is the final result of coarse registration. After obtaining the coarse registration result, the optimal transformation relationship between the two point clouds is obtained to optimize the registration result, that is, fine registration. The steps are as follows:

1) Use the point cloud $Y'(x_1, x_2, ..., x_n)$ and target point cloud $Y_{object}(x_1, x_2, ..., x_n)$ that have undergone coarse registration in the previous section as the point cloud dataset for fine registration;

2) Search for the nearest corresponding point in the target point cloud $Y_{object}(x_1, x_2, ..., x_n)$ for each point in the point cloud $Y'(x_1, x_2, ..., x_n)$ to be registered, and use the obtained point set as the initial corresponding point pair;

3) There may be significant errors in the corresponding transformation relationships in the initial corresponding point pairs, which can affect the final registration results. Direction vector thresholds are used to remove these highly erroneous corresponding points;

4) Calculate the rotation matrix and translation vector to minimize the mean square error between corresponding points;

Set the number of repetitions for the above process. When the number of repetitions exceeds 50, the point cloud to be registered is completely aligned with the target point cloud, achieving the effect of precise registration.

Rough registration is the rough matching of two unknown pose point cloud objects, while the fine matching

criterion is to minimize the spatial pose difference between the two point clouds based on rough registration. The process of point cloud registration is shown in Fig. 5.

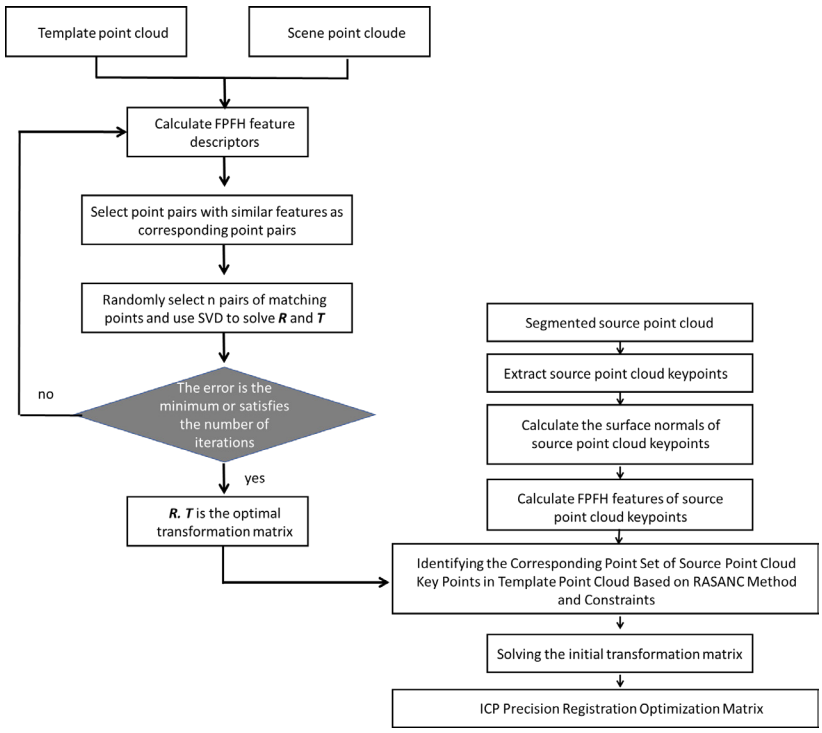

Fig. 5. Point cloud registration process

This article uses a sampling consistency strategy for coarse registration, and the registration pseudocode is as follows:

Solution steps
1: Calculate the Fast Point Feature Histograms (FPFH) feature descriptors for both the template point cloud and the scene point cloud separately;
2: Matching points in two point clouds using FPFH feature descriptors based on point clouds;
3: Randomly select n ($n \geq 3$) pairs of matching points;
4: Using singular value decomposition to solve for the rotation and displacement in this matching scenario;
5: Calculate the corresponding error at this time;
6: Repeat steps 3-5 until the conditions are met, and take the rotation and displacement corresponding to the minimum error as the final result.

Rough registration cannot obtain an accurate transformation matrix between two point clouds, and it essentially only utilizes a portion of the key points of the two point clouds, which can result in significant errors compared to the real transformation matrix. After transformation, the two point clouds only roughly overlap, so further optimization of the precise registration step is needed to obtain an accurate pose transformation relationship. The most commonly used algorithm for accurate point cloud registration is the Iterative Closest Point (ICP) algorithm, with the following pseudocode:

Solution steps
1: Initial pose transformation, assuming the initial rotation matrix obtained from coarse registration is R, the translation vector is M, and k is initialized to 0.
2: Find the nearest point set and search in the template point cloud; The nearest neighbors of each point form a point pair. If the distance between two points is greater than a certain threshold, it is ignored. Otherwise, it is added to the nearest point pair set.

3: Optimization of transformation parameters, estimating rigid transformation parameters based on the nearest point pair, and updating iteration errors:

$$\varepsilon_k = \frac{1}{m}\sum_{i=1}^{m}\left(q_{ki} - R_k p - t\right)^2$$

4: By rigidly transforming the matrix parameters, new point cloud data is obtained, while updating the transformation matrix and translation vector.

5: Repeat steps (2) - (4) above, and stop iterating when the number of iterations reaches the set value or the error is less than a certain threshold.

4.2 Pose Estimation

This article first determines the optimal grasping point of the part based on the pose obtained in the above chapters, and then plans the robot's grasping pose guided by the optimal grasping point [17].

According to Chapter 4.1, if the template point cloud Y (x_1, x_2, ..., x_n) is taken, the origin of the point cloud's coordinate system is O, and the vector y_i from the origin to each spatial point. By using coordinate transformation, the local coordinate system of the template point cloud is distributed with the maximum variance and the minimum covariance, that is, the covariance is 0 except for the diagonal values. Construct the covariance matrix of the template point cloud Y:

$$Cov = \frac{1}{n} B \cdot B^T. \tag{14}$$

Among them, B^T is the decentralized matrix, which decomposes the covariance matrix to obtain eigenvalues. The eigenvector corresponding to the maximum eigenvalue is the transformation formula of the coordinate axis. At this time, the template point cloud can be rotated to the origin of the workbench coordinate system through this transformation matrix. After converting the template point cloud to the origin of the workbench coordinate system, the maximum value of the x, y, z coordinate in the grasping direction can be calculated to obtain the grasping center. Assuming the determined grasping direction is the x-axis direction, the grasping center O_{zhua} is:

$$O_{zhua} = B^T\left(x_{\max} - x_{\min}, 0\right) + Cov. \tag{15}$$

This article uses a six degree of freedom robotic arm and a sampling based motion planning method to achieve obstacle avoidance planning and grasping of the robotic arm. By sampling the joint angle space, the complex modeling of environmental obstacle information in the angle space is avoided [18]. The planning process is as follows:

1) Initialize the joint angles of the industrial robot pose;

2) By randomly sampling the joint angle space and introducing a target bias strategy, O_{zhua} is the joint angle of the robotic arm obtained by reverse solving the target pose to be grasped. The probability of target bias set in this paper is 0.37. The schematic diagram of the reverse solving process is shown in Fig. 6:

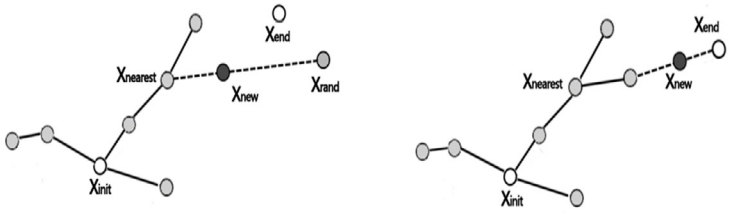

Fig. 6. Point cloud registration process

3) Select three sampling points from the initial path, use kinematic forward solution to obtain the corresponding pose of the node, and perform collision detection in the workspace. If the node experiences collision interference, add the new node to the random sampling array as the parent. However, if there is no interference, continue sampling.

4) Repeat steps 2-3. When the distance error between the point and the target node in the workspace is less than 1 mm and the attitude error is less than 0.01, obtain a path that can be planned and terminate the iteration.

The algorithm flowchart is shown in Fig. 7.

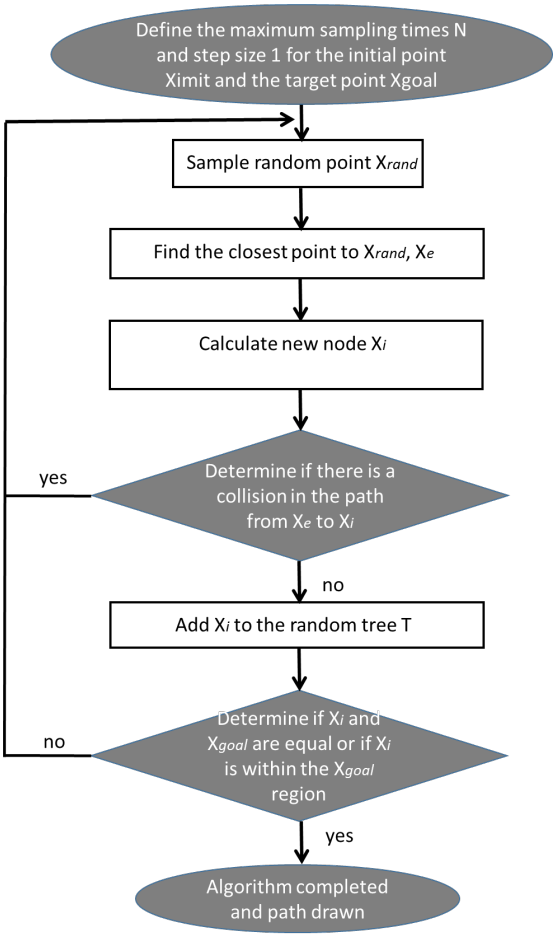

Fig. 7. Point cloud registration process

This section mainly describes the grasping strategy for stacked parts, omitting the point cloud feature extraction and processing steps. After the point cloud feature processing is completed, the part pose is determined through two processes: coarse registration and fine registration. Once the part pose is determined, the optimal grasping posture can be determined and the robot can be guided to complete the optimal path planning for part grasping. The planning method is described in the form of pseudocode.

5 Experiment and Result Analysis

This article uses stacked bearing outer rings for grasping experiments and conducts 8 registration experiments in actual scenarios. The registration experiment results are shown in Fig. 8.

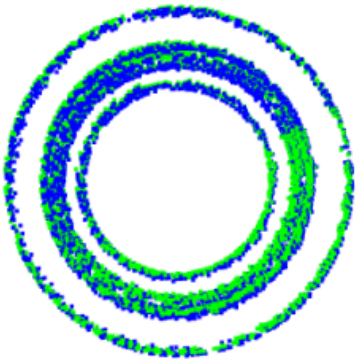

Fig. 8. Registration results

The statistical table of point cloud registration results is shown in Table 3.

Table 3. Point cloud registration results

Experimental group	Rough registration efficiency	Precision registration efficiency
The first time of the first group	0.027s	0.0478s
The second time of the first group	0.051s	0.059s
The third time of the first group	0.049s	0.624s
Second group, first time	9.903s	8.768s
Second group, second time	12.39s	12.089s
Second group, third time	4.986s	5.879s
The first time of the third group	28.974s	30.879s
Third group, second time	30.637s	29.768s
Third group, third time	4.982s	5.087s

After using the pose estimation method proposed in this article, the estimation efficiency has been significantly improved, and the estimation accuracy has also been greatly improved. The registration time has been greatly reduced compared to not using RANSAC to remove mismatched points. The final ICP scores for the registration of point clouds of the same type of parts in the first group and the experiment were all below 0.0025, indicating high registration accuracy.

Set conditions in the simulation environment ROBODK and obtain the trajectory planning path results as shown in Fig. 9.

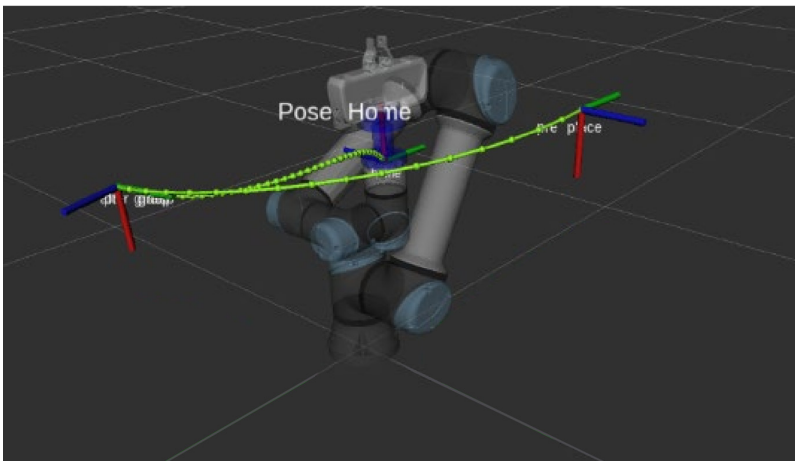

Fig. 9. Grasp trajectory planning

6 Conclusion

In the visual localization algorithm based on point to feature, this article provides a detailed introduction to the technical details and improvement measures during the implementation and implementation of the algorithm, which improves the recognition efficiency of the point to feature based visual localization method in practical use and enhances its performance in industrial parts. In robot motion planning, this article is based on the ROS robot operating system. Completed the modeling and kinematic forward and backward solutions of the UR 5 robotic arm. Avoiding the complex modeling of obstacles in the workspace in the joint angle space of the robotic arm, and discussing in detail the obstacle avoidance and grasping strategy of the robotic arm, the flexibility of the robotic arm operation is improved. In the comprehensive experiment of visual localization and grasping, this paper completed multiple sets of visual localization experiments for scattered and stacked scenes. A comprehensive comparison of segmentation driven visual localization methods shows that the visual localization strategy and method adopted in this paper have high accuracy and are easy to deploy, which has certain guiding significance for practical industrial production.

In terms of 3D visual localization, the visual localization method adopted in this article avoids the need for massive data based on learning methods. However, with the increase of object categories in the scene, object detection and instance segmentation strategies represented by deep learning can endow the scene with certain semantic information. By comprehensively using RGB image information and depth image information, the position of the grasping point can be estimated to avoid the dependence on the model in pose estimation and improve the robustness of the method.

7 Acknowledgement

Research on robot control system based on visual feedback (2023ZC015).

References

[1] J.-Q. Qin, Y.-L. Xie, Z. Jin, Y. Liu, Application of Industrial Robots and Enterprise Labor Cost Stickiness, Chinese Review of Financial Studies (5)(2023) 103-124+126.
[2] Z.-J. You, Research on development and application of industrial robot technology, China High and New Technology (16)(2023) 17-19.
[3] Y. Li, H.-Z. Guo, Research on Industrial Robot Grasping System Based on Machine Vision, Mechanical & Electrical Engineering Technology 52(10)(2023) 192-195.
[4] J.-X. Xu, H.-H. Zhang, L.-L. Chen, J.-L. Li, S.-R. Wen, Stacking Algorithm and Control Strategy of an Injection Molding Manipulator, Mechanical & Electrical Engineering Technology 52(12)(2023) 142-145.
[5] H.-F. Mai, X.-F. Yao, Pose Estimation of Scattered Stacked Bearing Rings Based on Point Cloud Deep Learning, Modular Machine Tool & Automatic Manufacturing Technique (11)(2023) 56-64.
[6] B. Liu, S.-F. Jin, Y.-W. Su, N. Yan, Y. Li, Robotic Unordered Sorting Method for Stacked Bobbin Yarns, Light Industry Machinery 40(6)(2022) 14-21.
[7] S.-J. Xu, K.-P. Li, J.-Q. Han, Y.-B. Meng, G.-H. Liu, Stacking Workpieces Sorting Algorithm Based on Multi-scale Feature Attention Yolact Network, Computer Measurement & Control 30(9)(2022) 184-192+200.
[8] J.-M. Zhai, L. Huang, 6D pose estimation and unordered picking of stacked cluttered objects, Journal of Harbin Institute of Technology 54(7)(2022) 136-142.
[9] X.-L. Zhu, G.-H. Cui, S.-X. Chen, L. Yang, Instance segmentation method based on improved Mask R−CNN for the stacked automobile parts, Journal of Shanghai University of Engineering Science 36(2)(2022) 168-175.
[10] J. Xu, N. Liu, D.-P. Li, L.-X. Lin, G. Wang, A Grasping Poses Detection Algorithm for Industrial Workpieces Based on Grasping Cluster and Collision Voxels, Robot 44(2)(2022) 153-166.
[11] Z.-L. Tong, L.-P. Sun, L.-L. Xu, X.-T. Lu, Research on the kinematics analysis model of flexible robotic arm of endoscopic robots, Modern Instruments & Medical Treatment 29(4)(2023) 46-52.
[12] Q. Cheng, H. Huang, J.-J. Xu, J.-H. Li, Y. Li, T. Zhang, An efficient robot hand-eye calibration method based on binocular vision and Halcon, Modern Electronics Technique 46(13)(2023) 35-42.
[13] S.-Y. Zhu, B.-J. Xiao, S.-Q. Huang, Irregular Part Grasping System of Manipulator Based on 3D Vision, Automobile Technology & Material (4)(2023) 60-67.
[14] L.-P. Qian, Z.-K. Shao, X.-L. Shen, Research on application of intelligent bin−picking technology of centrifugal pump impeller based on 3D machine vision, Modern Manufacturing Engineering (2)(2023) 27-35.

[15] H. Luo, J.-Y. Shang, R.-F. Li, J. Guo, N. Yang, C. Guo, A High-precision 3D Disordered Grabbing System for Industrial Robots, Navigation and Control 22(4)(2022) 106-116.
[16] M.-Y. Zhao, J.-J. Zhu, A Pose Detection Method for Robotic Arm Grasping based on 3D Point Cloud, Journal of Jilin Institute of Chemical Technology 40(11)(2023) 54-60.
[17] H.-W. Ma, N.-X. Sun, Y. Zhang, P. Wang, X.-G. Cao, J. Xia, Track planning of coal gangue sorting robot for dynamic target stable grasping, Journal of Mine Automation 48(4)(2022) 20-30.
[18] F. Wu, S.-J. Jing, X.-C. Lin, Grasping pose estimation method based on MaskR-CNN and key point extraction, Journal of Hefei University of Technology (Natural Science) 46(9)(2023) 1178-1184.