Journal of Computers Vol. 34 No. 6, December 2023, pp. 173-183
doi: 10.53106/199115992023123406014

Research on Control Strategy of Two Arm Collaborative Robot Based on Vision

Bing-Yan Wei[1,2*], Qian-Han Zhang[1], Xiao-Ying Wu[1]

[1] Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China
{wby83765, qianhan19881314, xiaoying88758}@163.com

[2] Intelligent Electronic Products Research and Development Center, Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China

Received 1 October 2023; Revised 1 November 2023; Accepted 13 December 2023

Abstract. Double arm robots are increasingly being used in automated production due to their higher work efficiency and better flexibility. This article focuses on the research of visual recognition based robots applied in automated production lines. Firstly, the visual system has been improved and image labeled to make it more accurate in identifying targets. Then, path planning has been carried out for the coordinated movement of both arms. Through the unconstrained collaborative planning method of both arms, the efficiency and robustness of path planning have been improved. Finally, experimental simulation has been conducted to verify the effectiveness of visual assistance conditions. The effectiveness of the dual arm planning method.

Keywords: vision, collaborative robots, planning

1 Introduction

With the development of intelligent manufacturing, industrial robots are increasingly playing a crucial role. Robots continuously improve the efficiency of intelligent manufacturing through advanced technologies such as big data, human-machine interaction, and CPS systems.

Collaborative robots are an excellent platform for implementing intelligent robots and are currently a research hotspot in the field of robotics. Collaborative robots can directly interact with people in the work area, and are lightweight, easy to install, and have very intuitive programming functions. They can be flexibly replaced with people in the production process.

The identification, sorting, and assembly of industrial parts are a common part of the industrial production process, with the aim of correctly classifying or selecting defective industrial parts from different types of workpieces. Traditional industrial parts sorting is achieved through manual operation, which has no advantages in sorting cost and sorting time compared to robot sorting. Moreover, robots can continuously work with high intensity. However, in practical applications, traditional robot systems mainly achieve fixed operations through offline programming or drag teaching, which cannot achieve automatic sorting function. The emergence of visual sorting robots has effectively solved the problem of insufficient intelligence, that is, by using visual systems to help robots obtain information about parts and environment, to assist robots in grasping and sorting target objects. However, current visual collaborative robots have the following defects:

1) Target recognition can quickly obtain the type of target object for sorting operations, while existing algorithms can only recognize simple and textured objects. On non textured objects, the recognition rate is low due to their lack of texture, complex structure, and other features.

2) The existing tracking algorithms can only track the plane information of the target object, that is, obtain the two-dimensional position of the object in the image, and cannot track the target pose information; There are a large number of similar objects on the sorting system of the assembly line, which is prone to recognition errors; The objects on the assembly line are constantly moving and the production environment is complex, making it easy to track and lose them.

3) During the trajectory planning process, the planning efficiency is low and there are extreme points along the route.

* Corresponding Author

Therefore, in response to the above issues, the work done in this article is as follows:

1) Firstly, the color image is segmented using a semantic segmentation model, and then the depth information of the object is obtained from the corresponding depth map, which is converted into a point cloud. Then, the nearest point cloud iteration algorithm for normal sampling of the corresponding points is used for accurate object pose registration. Thus achieving effective pose estimation for textureless and occluded objects in complex environments, providing necessary information for subsequent grasping;

2) According to the constraint relationship of trajectories in dual arm work, dual arm trajectory planning is divided into two categories: non constrained collaborative operation tasks and fully constrained collaborative operation tasks. Constraint equations are established for dual arm collaborative motion scenarios, and based on this, dual arm collaborative planning is carried out;

3) Build a simulation environment and conduct simulation experiments.

In summary, in order to express the above methods clearly, the article structure is as follows: Chapter 2 describes the relevant research results and provides some ideas, Chapter 3 discusses the work done for the visual part, Chapter 4 describes the path planning method for collaborative robots, Chapter 5 is the simulation section, and Chapter 6 is the conclusion section.

2 Related Work

Artemov utilizes a computer vision system to provide trajectory planning and motion control strategies for KUKA collaborative robot manipulators in dynamic environments, and adopts the potential field method as the real-time motion control algorithm for the robot. The simulation results demonstrate the improved performance of the control algorithm [1]. Runze Li designed a machine vision based mechanical fault image recognition system for automotive assembly collaborative robots, which utilizes machine vision technology to obtain mechanical fault image features and achieve mechanical fault image recognition for automotive assembly collaborative robots. The experimental results show that this method has good recognition performance and high recognition efficiency [2]. Shipei Li uses supervised learning algorithms to train and process the image information of the object to be captured, completing rapid recognition and localization of the captured target features; Then, the positioning information of the grasping object is transmitted to the ROS system through a serial communication protocol to control the robot to complete motion planning and obstacle avoidance motion for precise grasping [3]. Jiyong Fan collects surface images of pre formed bodies through image acquisition equipment, and then compensates for robot joint surface images through interpolation. Image processing is carried out using methods such as edge detection and two-dimensional discrete Fourier transform, and the carbon fiber workpiece is recognized using the discrete angle mean gray weave angle detection algorithm [4]. Deming Wang proposed an instance based segmentation network and iterative optimization method for the detection and pose estimation of common weakly textured and scattered objects in industry, which has reference significance for visual acquisition of weakly textured workpieces [5]. Jing Xia proposed a fast detection method for robot planar grasping pose based on cascaded convolutional neural networks for unknown irregular objects with arbitrary poses, and established a cascaded two-stage convolutional neural network model with coarse to fine pose. The methods and models have reference significance for this article [6].

3 Visual Processing

Visual processing includes two parts: visual image processing and visual system calibration. In order to achieve precise positioning and recognition of dynamic targets on the conveyor belt, the target image needs to be preprocessed first, and then Gaussian filtering is used to remove noise. The filtered image undergoes edge detection and feature extraction and recognition, and the image processing is completed. The calibration of the visual system consists of the calibration of the coordinate system of the collaborative robot tool, the calibration of camera internal parameters, and the calibration process of the hand eye of the collaborative robot.

3.1 Image Processing

This article briefly describes the image processing section below, with a focus on describing the improvement process of the collaborative robot vision system. This article uses an Intel RealSense D435 depth camera, which performs grayscale conversion on the image to improve the contrast between the target workpiece and the background [7]. The conversion formula is as follows:

$$h(x,y) = a \cdot f(x,y) + b. \tag{1}$$

In the formula, $h(x, y)$ represents the processed grayscale, $f(x, y)$ represents the grayscale of any point in the original image, and parameters a and b determine the contrast result. After the grayscale transformation, the image needs to be denoised. This article uses Gaussian filtering to denoise, converting the noise signal in the image into a high-frequency signal, and then assigning weights. The expression is as follows:

$$h(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2}$$

σ is the standard deviation, which determines the smoothness of the image. In this article, $\sigma = 1.0$ performs edge detection on the filtered image using the Roberts edge detection operator. The process is as follows:

$$G_x = f(x,y) - f(x+1, y+1). \tag{3}$$

$$G_y = f(x,y+1) - f(x+1, y). \tag{4}$$

$$G(x,y) \approx |G_x| + |G_y|. \tag{5}$$

$$G_x : \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{6}$$

$$G_y : \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{7}$$

Set threshold T to continuously compare $G(x, y)$ with T to extract edge points.

3.2 Feature Extraction

Using HSV color space for color comparison, extract pixel P components from the closed contour area of the image obtained in the previous process and perform discrimination:

$$\left(\left(P(i,j) \geq 0\right) \&\& \left(P(i,j) < 21\right)\right) \&\& \left(V(i,j) > 43\right) \&\& \left(S(i,j) > 46\right). \tag{8}$$

Count the number of pixels with red color features in the closed contour and set a threshold. If the number of pixels is greater than the set threshold, it is determined as red. The other color determination methods are the same, and the determination results are shown in Fig. 1.

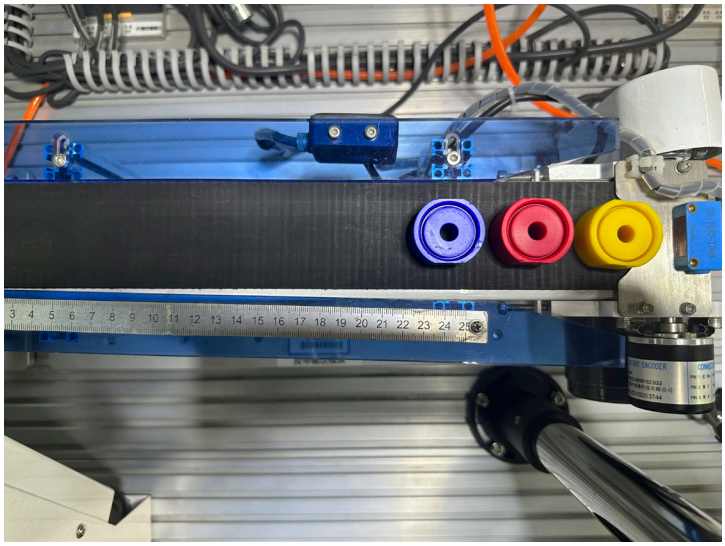

Fig. 1. Color judgment results

3.3 Vision System Calibration

Generally, robot fixtures are installed at the end of collaborative robots. After installing the tool, the coordinate system needs to be calibrated. Based on the size of the fixture, the height H of the home center is obtained, and the tool coordinates are obtained through coordinate transformation.

$$\begin{bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -H \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{9}$$

(x_T, y_T, z_T) represents the coordinates in the tool coordinate system, while (x, y, z) represents the coordinates in the collaborative robot base coordinate system. By converting the above equation, the coordinate conversion from the base coordinate system to the tool coordinate system can be completed, completing the calibration of the tool coordinate system.

The internal parameter matrix of the camera is represented as follows:

$$K \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{10}$$

In the formula, $f_x = af$, $f_y = \beta f$, and f are the focal lengths of the camera, the pixel coordinates are scaled α times on the u axis, β times on the v axis, c_x and c_y are the coordinates of the camera's optical axis in the pixel coordinate system, and K is the internal reference matrix. For the distortion of the camera lens, a comprehensive correction method is used for correction, and the processing process is as follows:

$$\begin{cases} \hat{x} = x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 xy + p_2\left(r^2 + 2x^2\right) \\ \hat{y} = y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1\left(r^2 + 2y^2\right) + 2p_2 xy \end{cases}. \tag{11}$$

Project the distorted point onto the pixel plane through the internal parameter matrix to obtain the correct position of the point on the image. The transformation method is as follows:

$$\begin{cases} u = f_x \hat{x} + c_x \\ v = f_y \hat{y} + c_y \end{cases}. \tag{12}$$

$\hat{x}$ and $\hat{y}$ are the distorted coordinates, $r^2 = x^2 + y^2$, k_1, k_2, k_3 represent the radial distortion coefficient, and p_1 and p_2 represent the tangential distortion coefficient. After calibrating the camera's internal reference coefficient, calibrate the camera, and the calibration process is shown in Fig. 2.

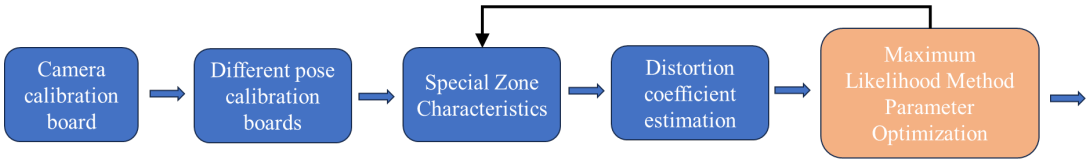

Fig. 2. Camera calibration process

By writing a script in the ROS [8] environment for camera calibration, the camera's internal parameter matrix can be obtained after calibration, which is represented as:

$$K = \begin{bmatrix} 1289.36 & 0 & 943.24 \\ 0 & 1238.76 & 523.89 \\ 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

The radial distortion coefficients are $k_1 = 0.017$, $k_2 = 1.28$, $k_3 = -4.92$, and the tangential distortion coefficients are $p_1 = -0.0035$, $p_2 = -0.0023$. The root mean square error (RMS) of the re projection is a comparison between the actual pixel coordinates and the calculated pixel coordinates, and the root mean square is calculated. The closer the result is to 0, the better the calibration effect is. The root mean square obtained is 0.173, which is within an acceptable range. The camera calibration is completed.

3.4 Collaborative Robot Calibration

According to the principle of eye to hand hand hand eye calibration, the OpenCV hand eye calibration function calibrateHandEye() is used in the ROS environment to program and complete eye to hand hand hand eye calibration. The hand eye calibration function calibrateHandEye() is used to solve problem $AX = XB$ [9].

Fix the camera directly above one side of the conveyor belt to ensure that the calibration board can be seen within the camera's field of view. Then, move the end of the collaborative robot 10 times and collect images of the calibration board 10 times. Each time the end of the collaborative robot is moved, the end position can be obtained based on the forward kinematics of the collaborative robot. After the calibration is completed, the hand eye calibration results will be printed on the terminal window of the Ubuntu system. Furthermore, the rotation matrix from the camera coordinate system to the cooperative robot base coordinate system is obtained, represented as:

$$R = \begin{bmatrix} -0.132 & 0.893 & -0.454 \\ 0.898 & 0.079 & -0.063 \\ -0.020 & -0.447 & -0.878 \end{bmatrix}. \tag{14}$$

The attitude angle obtained based on the rotation sequence of X-Y-Z fixed angles is:

$$RPY = [-150.24 \quad 1.12 \quad 89.63]. \tag{15}$$

The translation vector from the camera coordinate system to the cooperative robot base coordinate system is:

$$T = [0.668 \quad -0.321 \quad 0.689]. \tag{16}$$

After capturing the motion of the target object, the camera needs to calculate its position. Once the target object exceeds the working range of the cooperative robot, the robot will inevitably be unable to complete the grasping task. So this article needs to divide the conveyor belt according to the working range of the collaborative robot, as shown in Fig. 3. The detection of target objects is completed in the camera shooting area, and the collaborative robot completes the target grasping action in the grasping area. If the position of the target object calculated by the visual system exceeds the grasping area, wait for the conveyor belt to continue conveying the target object forward. After the target object enters the grasping area, the collaborative robot begins to execute the grasping action.

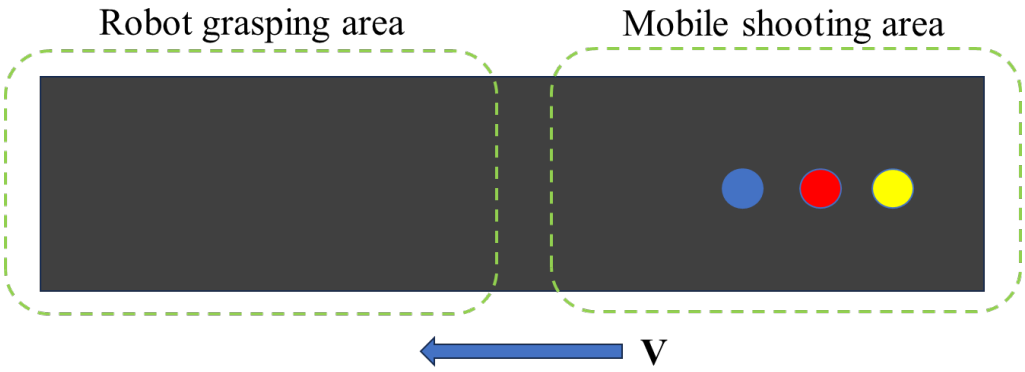

Fig. 3. Camera calibration process

4 Path Planning for Dual Arm Collaborative Robots

In the process of dual arm collaborative motion planning, the two robotic arms can be divided into two types of planning scenarios based on trajectory constraints, namely fully constrained collaborative motion planning and non constrained collaborative motion planning. The requirement for motion planning of a dual arm system in a fully constrained collaborative motion scenario is the highest. In addition to meeting constraints at both the starting and ending points, constraints cannot be broken during the motion process. If the two arms cannot always satisfy the constraint relationship during the fully constrained motion process, it may cause deformation of the object and even damage the mechanical arm.

4.1 Collaborative Robot Calibration

In the scenario of unconstrained cooperative motion of both arms, the main focus is on whether there are conflicts in the path trajectories of both arms. During the process of unconstrained cooperative motion of both arms, the motion planning plugin MoveIt provided by ROS is mainly used! Complete both arm motion planning. In ROS, collision detection is carried out through the structural model of the robotic arm. When a collision is detected with another robotic arm model in the running trajectory or with the environmental model, data indicating

planning failure is returned. When there is no conflict in the dual arm path trajectory, the dual arm work task can be completed. The experiment of simulating dual arm collision detection in ROS is shown in Fig. 4.

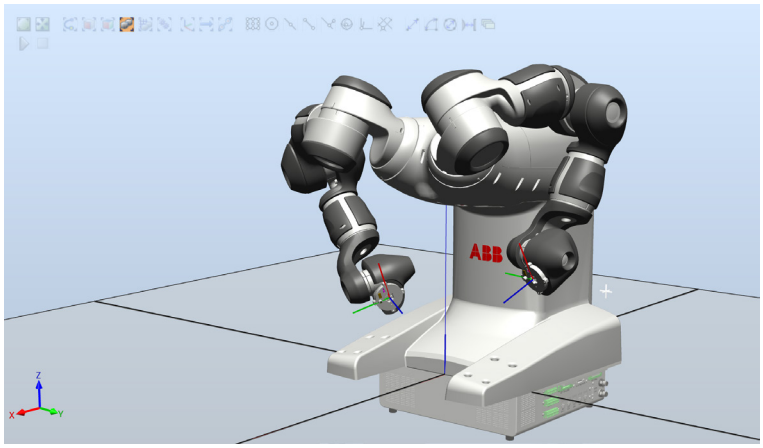

Fig. 4. Collision monitoring experiment

4.2 Dual Arm Fully Constrained Cooperative Motion

For two cooperative robotic arms, it is first necessary to obtain the constraint relationship between the position and velocity of the main arm end effector and the position and velocity of the slave arm end. After obtaining the constraint equation, the joint trajectory of the slave arm is immediately obtained. Due to the existence of constraint relationships throughout the entire motion process, if the trajectory of the main arm is modified online, the motion trajectory of the slave arm can be synchronously corrected through the constraint relationship.

Establish a coordinate system relationship in a dual arm scenario:

$$\left(x_m^t, y_m^t, z_m^t\right) \cdot \left(x_m^h, y_m^h, z_m^h\right) \cdot m = 0 \text{ and } m = 0,1,\cdots,6. \tag{17}$$

Based on the positional relationship between the two end coordinate systems $\left(x_6^t, y_6^t, z_6^t\right)$ and $\left(x_6^h, y_6^h, z_6^h\right)$, uniquely determined constraint conditions can be obtained, the constraint relationship is shown in Fig. 5.

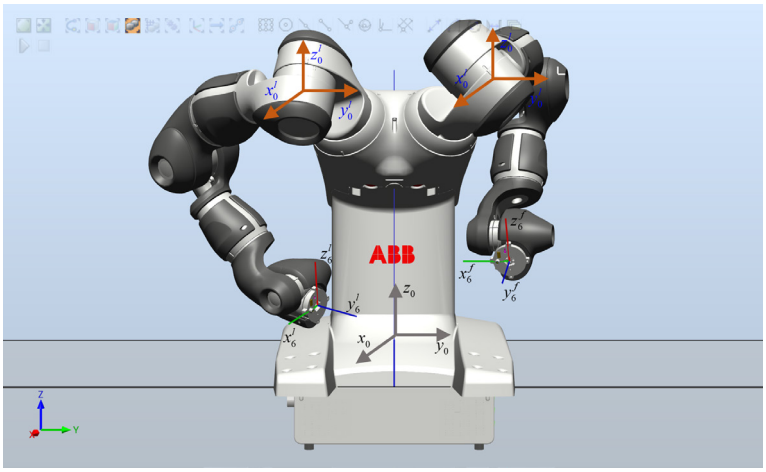

Fig. 5. Collision monitoring experiment

Let r^1 be the vector from the origin of the $\left(x_6^t, y_6^t, z_6^t\right)$ coordinate system to the origin of the $\left(x_6^h, y_6^h, z_6^h\right)$ coordinate system, and then obtain the position constraint relationship between the ends of the two arms as follows:

$$p\left(q^t\right) + Rot\left(q^t\right) r^1 - p\left(q^h\right) = 0. \tag{18}$$

The rotation constraint relationship is:

$$Rot\left(q^t\right)^T Rot\left(q^h\right) = U. \tag{19}$$

In the equation, q^t and q^h are the sequence of joint options for the left and right arms, respectively, and U is the 3×3 matrix, representing the rotation relationship between the ends of both arms. Based on speed constraint collaborative control of dual arm synchronization, establish dual arm differential kinematics. Let d_x, d_y, d_z represent the motion speed of the robot end along the spatial x, y, and z axes, and δ_x, δ_y, δ_z represent the optional angular velocity of the robot end around the x, y, and z axes, with the following relationship:

$$\begin{bmatrix} d_x \\ d_y \\ d_z \\ \beta_x \\ \beta_y \\ \beta_z \end{bmatrix} = \left[Jacobi\right] \begin{bmatrix} d\rho_1 \\ d\rho_2 \\ d\rho_3 \\ \beta\rho_4 \\ \beta\rho_5 \\ \beta\rho_6 \end{bmatrix}. \tag{20}$$

The Jacobian matrix is represented as $J(q) = [J_\lambda^T(q)\ J_\eta^T(q)]^T$, and the derivative changes are as follows:

$$\left[J_\omega\left(q^h\right) + L^1\left(q^h\right)\right] q^h - J_\omega\left(q^t\right) q^t = 0. \tag{21}$$

$$L^1 = \partial\left[Rot\left(q^t\right) r^1\right] / \partial q^t. \tag{22}$$

When two robotic arms simultaneously carry the same object and move, there is no relative motion at the ends of the two arms, that is, the velocity is consistent. Therefore, the detailed velocity constraint form between the two arms is:

$$\begin{cases} q^t = J^{-1}\left(q^t\right) \begin{bmatrix} J_\lambda\left(q^h\right) + L^1\left(q^h\right) \\ J_\omega\left(q^t\right) \end{bmatrix} \dot{q}^t \\ J^{-1}\left(q^t\right) = \begin{bmatrix} J_\lambda\left(q^t\right) \\ J_\omega\left(q^t\right) \end{bmatrix}^{-1} \end{cases}. \tag{23}$$

After the collaborative motion starts, real-time q^t is calculated based on the joint displacement of the arm, and continuously brought into formula 23 to calculate $\dot{q}^t$. When $\dot{q}^t$ changes, the trajectory of the dual robotic arms is continuously updated.

5 Grab Experiment and Result Analysis

The dual arm collaborative robot used in this article is IRB14000, which uses a dual arm robot to complete the installation of part modules. One arm holds the part base, and the other arm holds the electronic device installation. For ease of description, they are called the left and right robotic arms, respectively. The information required for environmental perception and object pose estimation is provided by Kinect v2. As shown in Fig. 6.

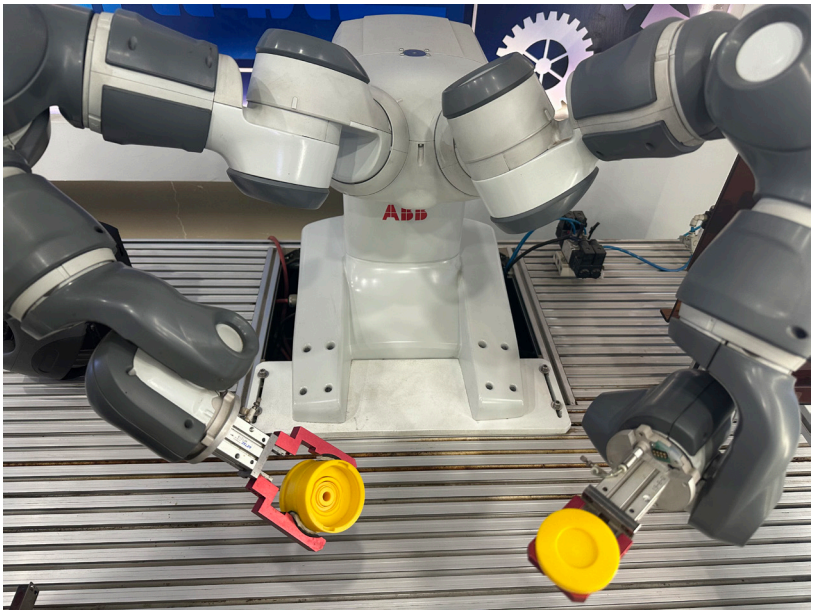

Fig. 6. Robot grabbing images

5.1 Simulation of Dual Arm Trajectory Planning

In order to perform dual arm trajectory planning in ROS, it is necessary to first establish a motion planning function package for the established dual arm URDF model, MoveIt! The configuration assistant (MoveIt! Setup Assistant) is provided to guide users in establishing motion plans through a GUI, as shown in Fig. 7.

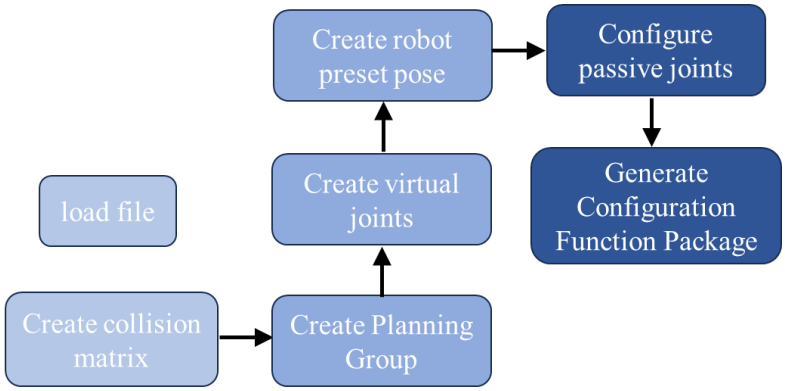

Fig. 7. Operational flowchart

By creating a planning group, trajectory planning can be carried out for multiple robots. The dual arm simulation scenario IRB14000 built in this article involves simultaneous planning of both arms, The trajectory planning simulation results are shown in Fig. 8.

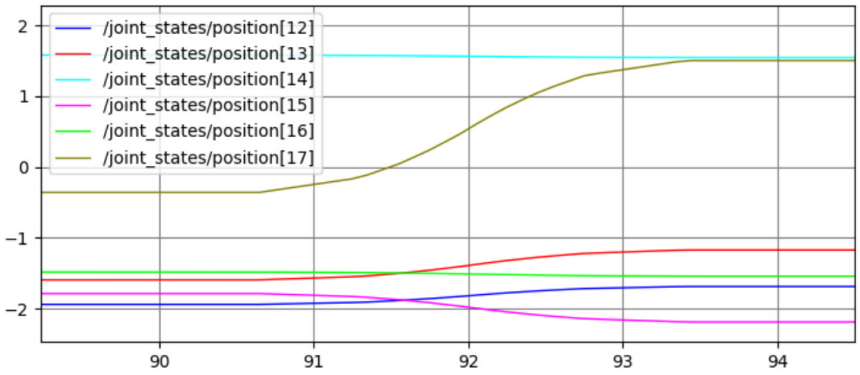

Fig. 8. Track planning simulation results

5.2 Grab Experiment

Conduct gripping experiments on an intelligent gripping system, placing the experimental objects randomly and randomly on the desktop. Use gestures to issue gripping commands to the robot. The robot will perform gripping based on pose estimation and path planning results, and close the end effector after reaching the designated position. The end effector will automatically determine the gripping force and opening angle based on force feedback. The experimental results are shown in Table 1.

Table 1. Capture Results

Bar clamp	Endstorholder	Mount2	Part1	Vase	Pinp Connector
4/5	3/5	2/5	5/5	0/5	0/5

From the figure, it can be seen that the algorithm proposed in this paper can plan a relatively smooth speed during the assembly process of mobile phone batteries in the laboratory, and all experimental indicators meet expectations, verifying the feasibility of the proposed method.

6 Conclusion

This article completes the effective pose estimation of objects with no texture and mutual occlusion in complex environments as a necessary guarantee for subsequent grasping work. Then, based on the trajectory constraint relationship in the work of dual arm robots, the dual arm trajectory planning is divided into two categories: unconstrained collaborative operation tasks and fully constrained collaborative operation tasks. A constraint equation is established in the dual arm collaborative motion scene to complete the dual arm collaborative planning, Finally, a simulation environment was built and the feasibility of the method was verified through experiments.

References

[1] K. Artemov, S. Kolyubin, Collaborative robot manipulator control in dynamic environment using computer vision system, Journal of Physics: Conference Series 1864(1)(2021) 012088.

[2] R.-Z. Li, S.-D. Qian, H.-N. Lu, Mechanical Fault Image Recognition System of Automobile Robot Based on Machine Vision, Machine Building & Automation 52(03)(2023) 189-192.
[3] S.-F. Li, J.-B. Han, B.-F. Lu, Z.-K. Qian, Research on Grasping Technology of Cooperative Robot Guided by Vision, Journal of Chongqing Technology and Business University (Natural Science Edition) 39(1)(2022) 42-48.
[4] J.-Y. Fan, Y.-J. Zhang, Z.-W. Wang, Z.-G. Yang, Visual detection of braiding angle on the surface of joint preforms of collaborative robot, Cotton Textile Technology 51(3)(2023) 1-7.
[5] D.-M. Wang, Y. Yan, G.-L. Zhou, Y.-Q. Li, C.-J. Liu, L.-M. Lin, Q.-J. Chen, 3D Vision-Based Picking System with Instance Segmentation Network and Iterative Optimization Method, Robot 41(5)(2019) 637-648.
[6] J. Xia, K. Qian, X.-D. Ma, H. Liu, Fast Planar Grasp Pose Detection for Robot Based on Cascaded Deep Convolutional Neural Networks, Robot 40(6)(2018) 794-802.
[7] Y.-P. Gong, Z.-Y. Zeng, F. Ye, Person re-identification method based on grayscale feature enhancement, Journal of Computer Applications 41(12)(2021) 3590-3595.
[8] W.-Y. Tao, J. Lv, Design and Implementation of the Automatic Grab System Based on ROS Robot, Journal of Fujian Computer 36(9)(2020) 14-19.
[9] H. Yan, X.-H. Cheng, Z.-M. Yang, C. Zhang, Calibration of Cooperative Welding Robot Based on Hand-Eye Relationship and Base Coordinate Relationship, Mechanical Engineering & Automation (4)(2020) 27-30.