MEKATRONIKA
JOURNAL OF MECHATRONICS AND INTELLIGENT MANUFACTURING
E-ISSN: 2637-0883
VOLUME 6, ISSUE 2, JULY, 2024, 20 – 38
DOI: https://doi.org/10.15282/mekatronika.v6i2.10654

RESEARCH ARTICLE

Dynamic Target Grasping Strategy of Industrial Robot Based on Machine Vision

Zhang XiaoYang[1,3], Muhammad Azmi Ayub[1,*], Fazlina Ahmat Ruslan[1], Sukarnur Che Abdullah[1] and Shuzlina Abdul-Rahman[2]

[1]College of Engineering, UiTM Shah Alam, 40450 Shah Alam Selangor, Malaysia
[2]College of Computing, Informatics and Media, UiTM Shah Alam, 40450 Shah Alam Selangor, Malaysia
[3]Hebei Institute of Mechanical and Electrical Technology, Xingtai, Hebei Province, China

ABSTRACT - Object grasping is the predominant application focus of robots, with machine vision technology playing a crucial role in enabling successful grasping. As research on machine vision technology advances, its initial application in capturing static targets has gradually expanded to include tracking and capturing moving targets. Detecting the target position is a common method for estimating and predicting its motion state, allowing for stable capture of moving targets. However, if the target's position changes unexpectedly due to interference, the original predicted position becomes invalid. Therefore, continuous localization and tracking of the target are necessary to assess deviations and approach the new target position in order to achieve successful capture. In this study, we utilize ROS as a platform to investigate dynamic grasping strategies based on visual feedback. We construct a dynamic grasping system using an UR5 industrial robot within ROS framework. The RealSense D435i camera is mounted at the end effector of the robot arm in an Eye-in-Hand configuration to obtain RGB-D image data representing the field of view at execution end point. By performing coordinate conversion, we acquire three-dimensional coordinates of objects in relation to base coordinate system of industrial robot. A visual feedback control strategy is designed to facilitate grasping operations on moving targets located on conveyor belts.When the target moved along the conveyor belt at a speed of 10mm/s, the camera realized accurate recognition of the object position, and the error was controlled within 1mm. At the same time, the industrial robot grasps and tracks the position error below 2mm to complete the target grasp.

ARTICLE HISTORY
Received : 6th May 2024
Revised : 23rd June 2024
Accepted : 16th Sept 2024
Published : 30th Sept 2024

KEYWORDS
Industrial Robot
Machine Vision
Target Detection
Dynamic Target Grasping

1.0 INTRODUCTION

Today, with the widespread adoption of industrial automation in production, robots have replaced human labor and are not only capable of freeing workers' hands, making product manufacturing more standardized and programmed, but also effectively reducing production costs, improving efficiency, and ensuring safety and quality. Among these applications, intelligent robots have found their most prominent role in industrial assembly line production [1]. The pick-and-place action is a crucial technology in assembly line operations. In real working conditions, pick-and-place tasks such as parts assembly, express sorting, food packaging, and palletizing are complex and intricate processes that require a high level of automation. Traditional teaching methods for workpiece picking and placing follow fixed trajectories which limit adaptability to different environments and lack flexibility. Objects scattered on conveyor belts often pose challenges in determining target points; thus traditional teaching modes fail to meet actual requirements [2]. Consequently, machine vision technology has been introduced into the academic community over recent years for intelligent object identification and positioning on industrial assembly lines. Due to its high autonomy and intelligence levels it has become a global research hotspot within the field of intelligent robotics. Machine vision can be considered as the "eyes" for robots by providing them with environmental information necessary for decision-making processes. Collected signals are processed enabling discrimination capabilities along with intelligent decision-making abilities that meet both quality standards as well as efficiency requirements within industrial manufacturing [3-4].

The integration of machine vision technology in the field of industrial assembly line picking and placing sorting involves equipping robots with various image sensors to enable the application of machine vision technology for simultaneous determination of target characteristics and types. This allows real-time tracking and acquisition of target locations on the conveyor belt, facilitating intelligent identification and sorting on the industrial assembly line. The introduction of machine vision technology enhances the environmental adaptability of robots during industrial assembly line operations, resulting in improved accuracy and efficiency [5-6].

Industrial robots possess multiple degrees of freedom and complex control systems. To address challenges related to difficult design, poor portability, and algorithm implementation issues associated with traditional robot arm control systems, TOSEPH [7] et al. developed a widely accepted Robot Operating System (ROS). ROS provides a unified

***CORRESPONDING AUTHOR | Muhammad Azmi Ayub** | ✉ muhammadayub@salam.uitm.edu.my

software platform for studying obstacle avoidance strategies in industrial robots [8]. This platform enables group design of control code as well as modular design of machine systems, thereby enhancing robot design efficiency [9-10].

In this study, we construct a dynamic grasping platform for UR5 industrial robot based on visual feedback within the ROS framework to achieve rapid recognition of target objects. Additionally, we calculate the offset at the end point of the robot arm using visual ranging techniques. A visual feedback control strategy is designed to approximate fixation during grasping operations involving moving targets on conveyor belts.

2.0 MODEL AND METHOD

Based on the advantages of ROS robot operating system, such as free and open source, and easy to extend the code, the author of this paper built a dynamic grasping platform with ROS-Melodic in Ubuntu18.04. Core components include: conveyor belt, RealSense D435i camera, UR5 industrial robot, robotiq_85_gripper and so on. The camera is connected to the end of the robot arm in an Eye-in-Hand mode to obtain the field of view of the execution end. The UR5 industrial robot is fixed on the side of the conveyor belt running direction as shown in Figure 1.

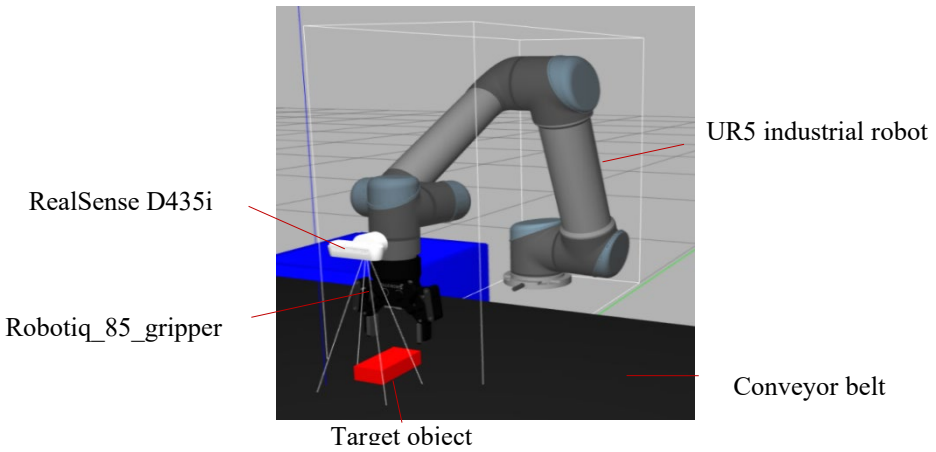


Figure 1. UR5 dynamic grasping system of industrial robot

2.1 Robot Operating System ROS

We use ROS as a platform to operate the UR5 robot, connecting all the steps of the operation. ROS systems are now the standard in robotics, working primarily on Linux and being the default platform for UR5 robots. Communication between all nodes is done through a dedicated ROS topic. A node running a reverse kinematics algorithm reads a message specifying the desired end-effector pose and calculates the desired joint angular position, and these control elements are then published in separate topics that are read by a low-level controller for each arm of the robot. The whole process runs at a frequency of 100Hz, which can be considered real-time for the actual situation of robot movement.

2.2 Robotic Arm Configuration

On ROS platform, SolidWorks is used to establish 6-DOF UR5 robot model and robotiq_85_gripper model. sw_urdf_exporter plug-in is used to determine the connection relationship and coordinate relationship of each joint component of the robot arm. Automatically generate URDF files and launch files for display. The Unified Robot Description Format (URDF) is a file that describes a robot model in XML format for use by the ROS platform. Create a MoveIt configuration function package, MoveIt configuration assistant makes a series of configurations for The robot arm model and completes the compilation, and uses KDL(The Kinematics and Dynamics Library) to solve the forward and inverse kinematics. Complete the configuration, visualization and simulation of the robot. The detailed steps are shown in Figure 2.

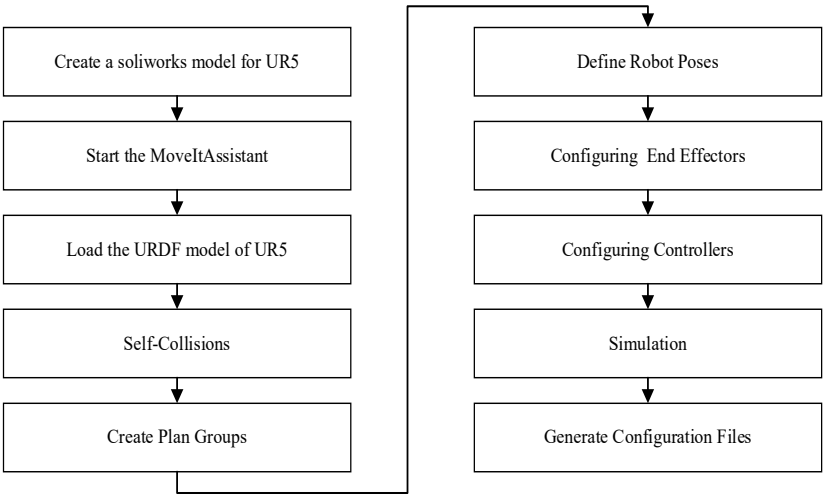


Figure 2. Configuration process of Moveit manipulator

2.3 Kinematic Analysis of Robotic Arm

The kinematics analysis of the UR5 industrial robot is carried out, and the structure diagram of the UR5 industrial robot is shown in Figure 3 and Figure 4.

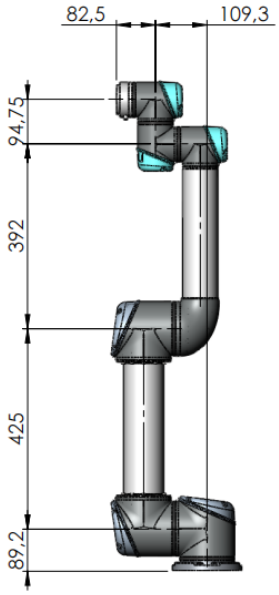


Figure 3. UR5 dimensional drawing

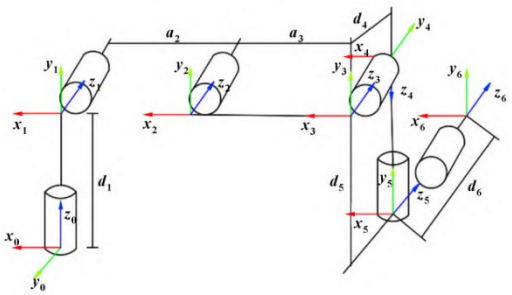


Figure 4. UR5 industrial robot structure diagram

The forward kinematics of a manipulator arm is a process of obtaining the pose of the end-effector by means of coordinate transformation. The D-H coordinate system of the UR5 industrial robot model represents the pose transformation relationship between each moving joint, and its D-H parameters are shown in Table 1.

Table 1. D-H parameters of UR5 industrial robot

Joint axis i	di(mm)	αi(mm)	αi(°)	Joint angle θi and scope(°)
1	89.459	0	90	θ1(-170～170)
2	0	-425	0	θ2(-150～150)
3	0	-392.25	0	θ3(-115～115)
4	109.15	0	90	θ4(-150～150)
5	94.65	0	-90	θ5(-150～150)
6	82.3	0	0	θ6(-300～300)

After determining the parameters and coordinate system of each link of the manipulator, the D-H parameter method is used to determine the transformation matrix of each link. The final transformation matrix can be obtained by multiplying the transformation matrix of each connecting rod.The transformation matrix of the i-1 link is shown Equation (1).

$$
\begin{aligned}
{}^{i-1}_{i}T &= Rot_{z_{i-1}}(\theta_i)Trans_{z_{i-1}}(d_i)Rot_{x_i}(\alpha_i)Trans_{x_i}(a_i) \\
&= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{1}
$$

$$
{}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

Substituting the above DH parameters into the above formula, the transformation matrix can be obtained as shown in Equation (3) :

$$
\begin{aligned}
{}^{0}_{1}T &= \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{3}_{4}T &= \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & 0 \\ \sin\theta_4 & 0 & -\cos\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{4}_{5}T = \begin{bmatrix} \cos\theta_5 & 0 & -\sin\theta_5 & 0 \\ \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{5}_{6}T = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3}
$$

Thus, the coordinate conversion equation of UR5 industrial robot can be obtained, as shown in Equation (4):

$$
T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}
$$

Where: n, o, a represent the attitude of the end effector relative to the base coordinate; p represents the position of the end effector with respect to the base coordinates[11-12].

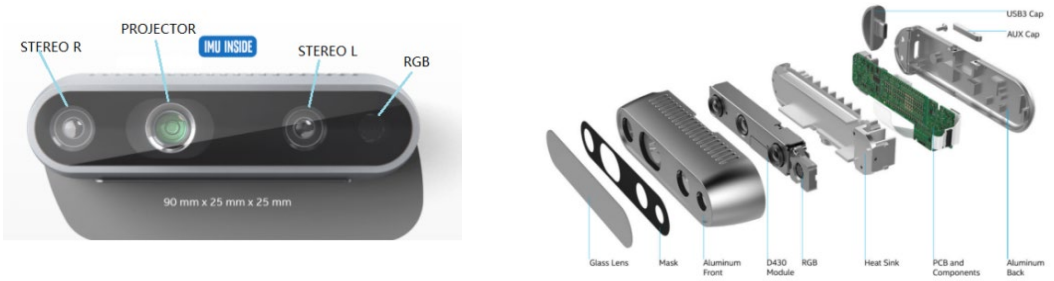
Similarly, if the pose T of the end coordinate system is known, the rotation angles of each joint (θ 1- θ 6) can be solved using linear algebra according to Equation(4), which is called the solution of the inverse kinematics of the robot.

2.4 Target Identification and Spatial Positioning

The RealSense D435i is a stereoscopic depth Camera, shown in Figure 5, that integrates two IR Stereo cameras, an IR Projector, and a Color Camera. The stereo depth camera system mainly consists of two parts: vision processor D4 and depth module. The host processor is connected to USB 2.0 or USB 3.1 Gen 1. The vision processor D4 is located on the main processor board, and the RGB color sensor data is sent to the vision processor D4 through the color image Signal processor (ISP) on the main processor board and the D4 board. The RealSense D435i offers a complete depth camera

module that integrates a vision processor, a stereo depth module, an RGB sensor, and a color image signal processing module. The depth module uses a left and right imager for stereo vision, an optional infrared laser emitter, and an RGB color sensor.

The RealSense D435i camera uses classic binocular vision to measure depth. Although it has an infrared projector, it does not use infrared reflection ranging. Its role is only to project invisible fixed infrared texture styles, improve the depth calculation accuracy in the environment where the texture is not obvious (such as white walls), and assist binocular visual ranging. The left and right cameras feed the image data into the built-in depth processor, where the depth value of each pixel is calculated based on the principle of binocular ranging.



(a) (b)

Figure 5 (a) RealSense D435i camera **(b)** explode diagram

2.4.1 Principle of Binocular Stereo Ranging

By calculating the parallax of the two images, the distance of the scene in front (the range captured by the image) is measured directly, without judging what type of obstacles appear in front. Therefore, for any type of obstacle, the necessary early warning or braking can be carried out according to the change of distance information. The principle of binocular cameras is similar to that of human eyes. The human eye is able to perceive the distance of objects because of the difference between the images presented by the two eyes on the same object, also known as "parallax". The farther away the object, the smaller the parallax; Conversely, the greater the parallax. The magnitude of the parallax corresponds to the distance between the object and the eye[13-14].

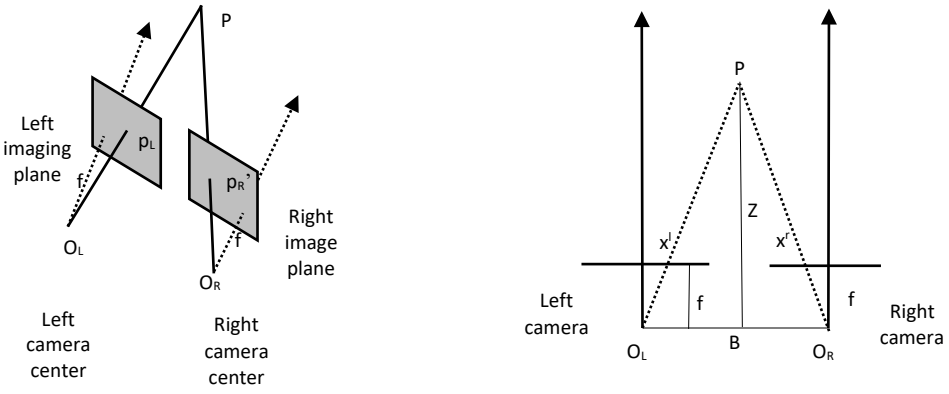
In Figure 6, $\mathbf{O_L}$ and $\mathbf{O_R}$ are the optical centers of the left and right cameras respectively, $\mathbf{p_L}$ and $\mathbf{p_R}$ are the image points formed by the target **P** on the imaging plane of the left and right cameras respectively, f is the focal length of the camera, **Z** is the distance between the target and the observation position, and **B** is the horizontal distance placed by the two cameras, which is called the baseline. Suppose that the coordinates of the image point $\mathbf{p_L}$ formed by the target point **P** on the left image plane are $\mathbf{(x_l,y_l)}$, and the coordinates of the image point $\mathbf{p_R}$ formed on the right image plane are $\mathbf{(x_r, y_r)}$. Because the two cameras are arranged on the same horizontal line and the optical axes are parallel to each other, the projection points on the left and right imaging planes should be aligned on the same line, parallax can be obtained $\mathbf{D=x_l-x_r}$. According to the similar triangle principle, it is easy to obtain, as shown in Equation (5):

$$\frac{B-(x^l-x^r)}{Z-f}=\frac{B}{Z} \Rightarrow Z=\frac{fB}{(x^l-x^r)} \Rightarrow Z=\frac{fB}{D} \tag{5}$$

From this, the distance between the object and the camera coordinate system is calculated, and the depth value **Z** of the scene .

At the same time, the X and Y coordinates of point **P** in the camera coordinate system can be calculated, as shown in Equation (6):

$$X=\frac{x-c_x}{f}Z \qquad \mathrm{Y}=\frac{\mathrm{y}-c_y}{f}Z \tag{6}$$



(a) (b)

Figure 6 (a) Standard binocular vision system model **(b)** Top view

2.4.2 Obtain the three-dimensional coordinate value of the target world coordinate system

Based on the images captured by the RGB camera, the image coordinate system is converted to the world coordinate system. The conversion principle is shown in Figure 7.

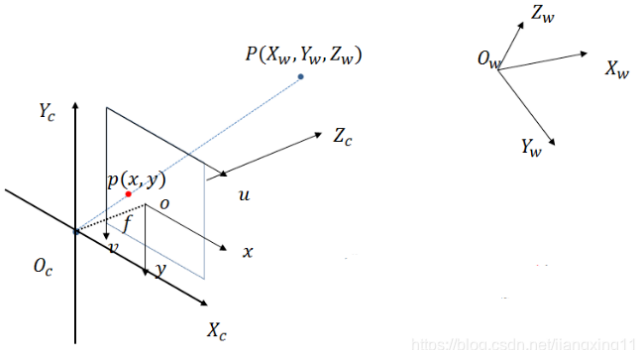


Figure 7. Schematic diagram of image coordinate conversion world coordinate

As shown in Figure 7, according to the principle of pinhole imaging, the transformation relationship between a certain point **P(X_w, Y_w, Z_w)** in the world coordinate system and its projection point **p(u, v)** on the image plane is shown in Equation (7) :

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_1 M_2 X_w \tag{7}$$

Where :$\mathbf{f_x}$ and $\mathbf{f_y}$ are the effective focal lengths of u axis and v axis respectively; $\mathbf{M_1}$ and $\mathbf{M_2}$ are camera internal parameters matrix and external parameters matrix respectively. **R** is the rotation matrix; **T** is the translation matrix[15].

Through the hand-eye calibration of the camera, the internal parameter of the camera and the external parameter matrix of the conversion between the robot arm and the camera are obtained, and the conversion from the two-dimensional image to the three-dimensional world coordinate system is finally completed, and the pose data of the feature points of the target is solved[16].

2.4.3 Acquisition of target feature points

In machine vision-based robot arm grasping systems, the primary objective is the precise identification and localization of the object block, a critical determinant of the ultimate success rate of grasping. To accurately derive the coordinates of the target within three-dimensional space, the principal process of target object recognition and positioning is delineated in Figure 8, subsequently accompanied by a comprehensive analysis.

Within the visual servo control system, the image processing procedure entails extracting feature data from images captured by the industrial camera, subsequently computing motion data for the target based on these features, and providing it as input to the robot motion control system. Given that the speed and temporal consistency of image

processing directly influence the efficacy of robot motion control, efforts are directed towards mitigating image processing time discrepancies. This study initially preprocesses camera-captured images, conducts feature extraction, target recognition, and positioning, then transforms the target feature positions into the world coordinate system before serving as input to the robot motion controller.

Image preprocessing plays a pivotal role in machine vision systems. Unlike increasing image information, its objective is to diminish irrelevant data within images, enhance image quality, attenuate non-essential information for processing or analysis, and amplify pertinent analytical features. By mitigating the effects of lens distortion, lighting variations, noise, and other factors, preprocessing ensures that images are in an optimal state for subsequent feature extraction. In this research, the image preprocessing pipeline encompasses three key steps: image smoothing, image enhancement, and image segmentation.

In industrial settings, the acquisition of images invariably entails encountering noise. These disturbances often manifest as outliers in pixel values, resulting in bright or dark spots that degrade image quality. Such anomalies can impede subsequent image processing tasks, including filtering, edge detection, and feature recognition, potentially leading to inaccuracies in processing outcomes. Clear delineation of edge outlines is crucial when capturing vector data of workpieces. Image smoothing serves the purpose of eliminating interference within the image, rendering its salient features more discernible, and enhancing the signal-to-noise ratio and overall image quality while preserving image integrity.

The Gaussian filter is filtered through the Gaussian filter, which converts the noise signal into high frequency signal, uses the Gaussian function to determine the weight in the template, and uses linear convolution to complete the denoising. The Gaussian template gradually decreases from the center to the periphery. Its expression is:

$$g(x,y)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{8}$$

In the process of Gaussian filtering, the pixels in the filter kernel are convolved with the convolution kernel, and the result is assigned to the core pixel of the filter kernel.

Gaussian blur is very effective in removing Gaussian noise from images.

Opencv encapsulates the Gaussian filtering as the GaussianBlur( ) method.

dst = cv2.GaussianBlur(img, ksize, sigmaX, sigmaY)

Parameter img is the original image, parameter ksize is the size of filter kernel, and sigmaX and sigmaY represent the standard deviation in the X and Y directions respectively.

Only binarization images of all target objects can be obtained through image preprocessing. To recognize and locate target objects, color features and geometric center pixel coordinates of each target object need to be further obtained. Therefore, HSV threshold segmentation, contour extraction, template matching and other algorithms are also needed to complete the feature extraction operation.

(a) HSV color segmentation

In this research, the target object can be preliminarily recognized and classified by color features. Compared with the common RGB color space, the specific color information of the object is not intuitive, HSV (Hue, Saturation, Value) space is a visual color model oriented to visual perception, with better recognition, so this experiment adopts HSV color segmentation algorithm to segment the color of the target object. This is shown in Figure 9.

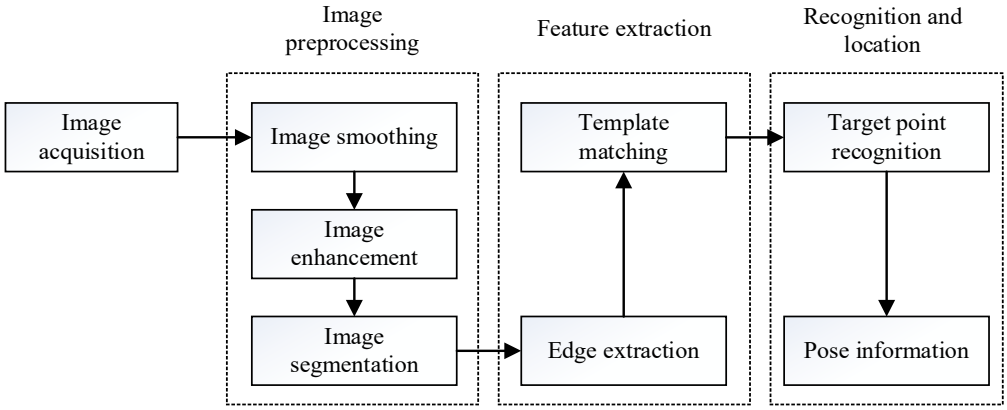
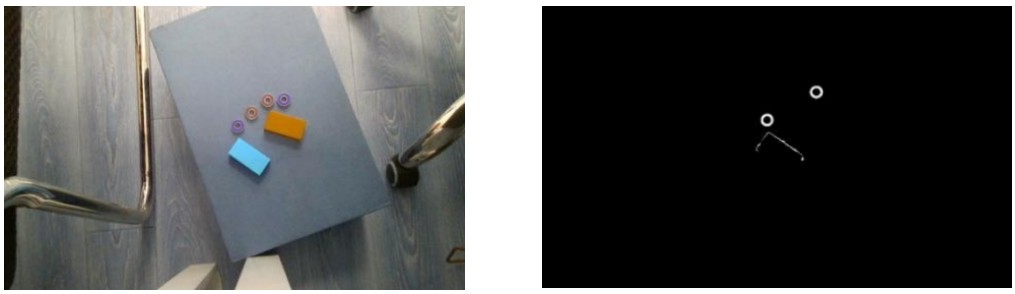


Figure 8. Image processing and recognition flow chart



(a) Original image (b) Converted image

Figure 9. Color extraction purple objects

(b) Morphological processing

Morphological processing encompasses four operators: dilation, erosion, closing operation, and opening operation. These operators are instrumental in addressing various image processing challenges such as noise suppression, feature extraction, edge detection, and image segmentation.

Binary processing results in uneven image edges, holes and noise areas. In order to further optimize the image, morphological processing algorithm is used, which mainly includes two operations, corrosion and expansion, also known as maximum and minimum filters. Expansion is the operation of finding the local maximum, and the expansion operation is used to integrate the background point into the object and fill the hole. Corrosion is the operation of local minimization. Corrosion is used to eliminate boundary points and shrink boundaries. In order to reduce the impact on the image contour, corrosion operation can be performed first, followed by expansion operation, or expansion operation can be performed first, followed by erosion operation, corresponding to the opening and closing operation of morphology respectively.

Bloat: The bloat operation can spread the white area (the black area shrinks), which is mostly used when you want to connect the white pixels together, but the white part will become larger;

Corrosion: Corrosion, as opposed to expansion, can spread the black area (the white area shrinks);

Closed operation: first expansion and then corrosion, the broken part of the white area can be connected (the black connected part is disconnected), while not affecting its own size;

Open operation: corrosion after expansion, open operation is opposite to closed operation;

The filter kernel includes: rectangle, ellipse, diamond; The filter kernel size can be set by width and height.

The OpenCV image dilation function is encapsulated as the dilate() method.

```
dilate(img, dilated_cv, Mat())
```

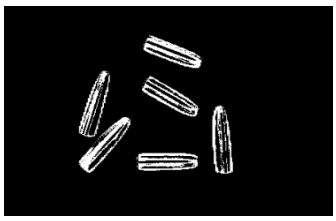
Where img represents the input image, dilated_cv represents the output image, and Mat() represents a default kernel, looking for the maximum in the 3×3 range.

The OpenCV image bloat function is encapsulated as the erode() method.

```
Mat eroded_cv;
erode(img, eroded_cv, Mat());
```

Where img represents the input image, eroded_cv represents the output image, and Mat() represents a default kernel, looking for a minimum value in the 3×3 range.

The effect diagram of each operator is shown in Figure 10.

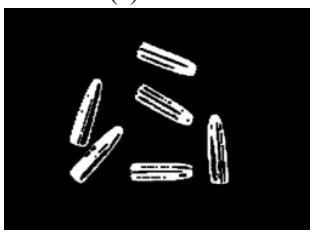


(a) Original image

(b) corrosion

(c) dilate

(d) Open operation



(e) Closed operation

Figure 10. Each operator application effect diagram

(c) Edge extraction

To obtain the pixel coordinates of the geometric center of each target object, the contour detection function (cv2.findContours()) from the OpenCV vision development library is utilized. Given that the shape features of the target object post-binarization exhibit regular geometric patterns, contour extraction is feasible. Following image preprocessing, contours and pixel coordinates of the target image are extracted. For a more visually comprehensible depiction, the contours are superimposed on the original image, as depicted in Figure 11. The image illustrates contour extraction of all red target objects, post-HSV color segmentation operation.

(a) Original image

(b) Converted image

Figure 11. Edge extraction

(d) Template matching

Following the procedure, while the vision system acquires the central pixel coordinates of each target object, the grasping system still struggles to discern distinguishing features among them. Therefore, after employing the contour recognition function from the OpenCV vision library to detect contours, it becomes imperative to utilize the template matching algorithm for further refinement of the preprocessed targets.

The fundamental principle of the template matching algorithm involves employing a reference template for comparison to locate similar templates within the source image. During the matching process, the source image dataset is indexed with templates from the library (where target objects of various shapes are pre-photographed and used as templates after binarization processing). Comparison results between the template and the image are stored in a matrix during the matching process. Subsequently, the scoring model is utilized to determine the optimal position of template estimation within the dataset.

In this research, the shape features of the target objects are distinctly discernible, rendering the template matching method highly effective in recognition. The template matching results, based on contour features, are illustrated in Figure 13.

Figure 12. Training template

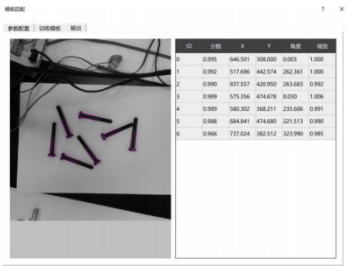


Figure 13. Template matching

Build OpenCV workspace based on ROS platform under Linux system. OpenCV is an open source cross-platform computer vision library, supporting the use of Linux, Windows and other platforms, it is efficient, lightweight and other advantages, and can achieve many aspects of computer vision and image processing functions [17].

OpenCV can be used to control the robot to recognize the location of the target and process the estimated location. The UR5 robot system publishes camera information as a subject in ROS, a feature that helps build ROS nodes without delay.

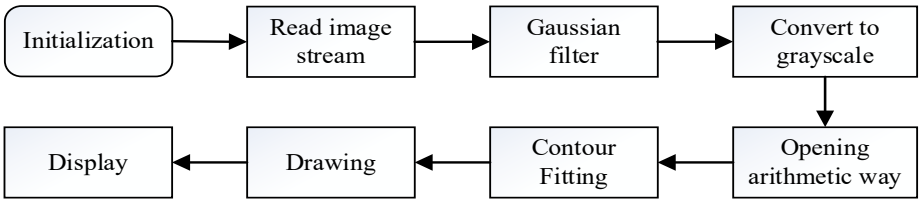


Figure 14. OpenCV image processing flow

By calling the OpenCV library, python programs are written, and the program implementation process is shown in Figure 14. Finally, the image is displayed in real time, the maximum contour identified in the image is framed, and the center point is drawn to get the rotation Angle of the maximum contour object. The result is shown in Figure 15.

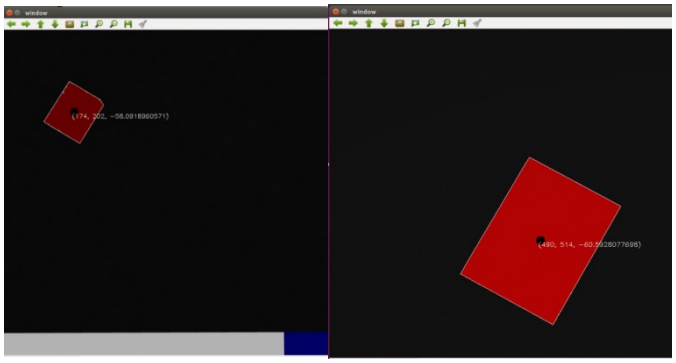


Figure 15. Result of the target pose

2.4.4 Target pose calculation

After obtaining the pixel coordinates of the center point of the target object in the image, the OpenCV function library uses **aligned_depth_frame.get_distance()** and **rs.rs2_deproject_pixel_to_point()** to obtain the three-dimensional coordinates of the camera coordinate system. Through the external parameter matrix (including rotation matrix R and

translation matrix T, obtained by hand-eye visual calibration of the robot), the three-dimensional coordinate value of the robot coordinate system of the feature target point is obtained after transformation.

The program flow is shown in Figure 16:

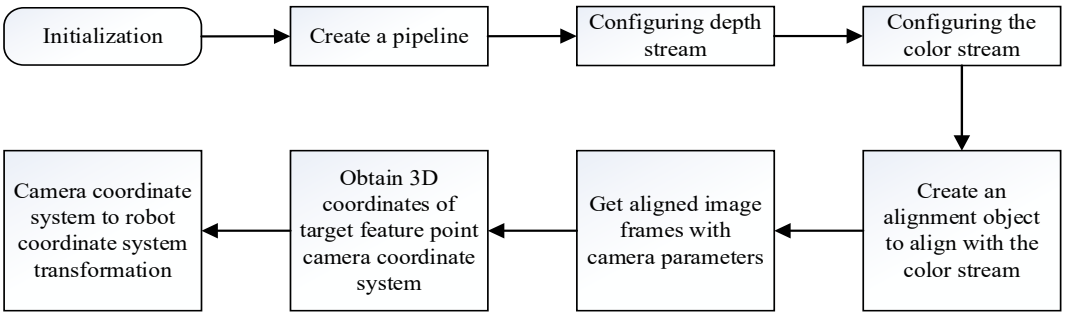


Figure 16. Flowchart of OpenCV obtaining 3D coordinates

Through program processing, the three-dimensional coordinate values of the robot coordinate system with fixed object feature points are calculated, and the data in Table 2 are obtained. The camera is fixed to the position P, and the target position A, B and C are changed to obtain the three-dimensional coordinates and rotation Angle; Fixed the target position A, changed the camera position P, Q, R, to obtain three-dimensional coordinates and rotation Angle. The feature points of the target object and the transformation of the robot coordinate system have been achieved.

Table 2. The pose data of the object relative to the base coordinate system

Location Camera	Location Object	Position object			Rotation angle
		X	Y	Z	θ
P	A	137.204	201.147	134.588	58
P	B	116.919	201.519	133.567	32
P	C	87.0767	201.337	133.149	32
P	A	137.204	201.147	134.588	58
Q	A	137.593	200.782	132.337	58
R	A	136.991	199.923	133.874	58

2.5. Robot target grasping based on machine vision

The proposed system, as shown in Figure 17, is mainly composed of three stages, which helps to improve the accuracy of the robot to grasp objects. Stage 1: Visual recognition via the arm attached Realsense 435i camera. Firstly, the object is identified, the pixel coordinates and rotation Angle of the center point of the object are obtained in the image, and then the three-dimensional coordinate values of the robot base coordinate system are converted by internal parameter and external parameter matrix.

Stage 2: After identifying the object to be grasped, the robot first solves the inverse kinematics, calculates the motion trajectories of each joint according to the coordinate positioning of the object to be grasped, and realizes that the end of the actuator reaches directly above the object.

Stage 3: Build an automated system to achieve target recognition and positioning, space movement of the robot arm and orderly arrangement of targets.

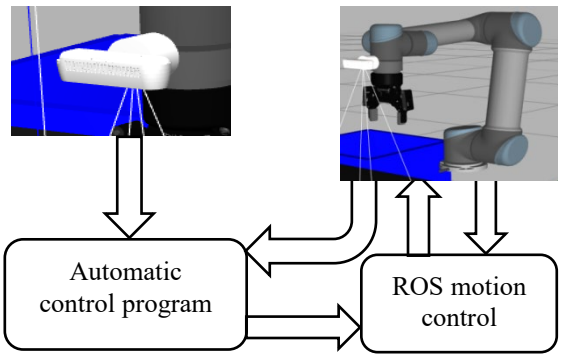


Figure 17. Control system diagram

2.6. PID-based dynamic target tracking and capturing algorithm

In the process of tracking and grasping the target of the robot arm, after determining the grasping position, the position between the robot arm and the target of the conveyor belt should be constantly adjusted and reduced.The schematic diagram of robot control and tracking is shown in Figure 18.

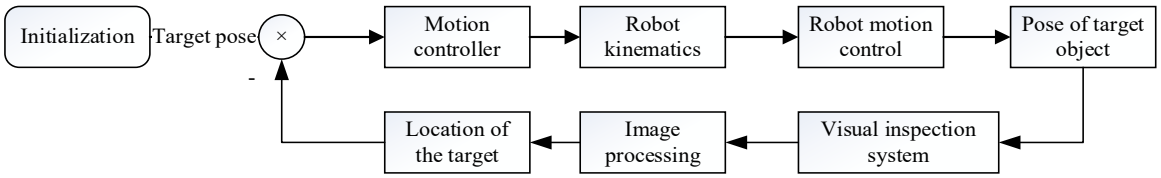


Figure 18. Schematic diagram of robot control and tracking

The conveyor belt is arranged in the XOY plane of the coordinate system of the robot arm, and the dynamic grasping process only requires real-time tracking in the XY orientation.

The dynamic tracking and grasping based on PID form realizes the grasping of objects by chasing, which is a synchronous tracking method of robots [18]. The target object runs to the edge of the working space of the robot arm, and after entering the camera, the vision system processes the acquired image and identifies the target, and then determines the three-dimensional coordinates of the captured position. In the process of moving, the orientation of the end of the manipulator is adjusted in real time to be consistent with the current position of the dynamic target by refreshing the position of the high-frequency target. Currently, the PID control algorithm is used in the robot position mode to adjust the position between the two, so that the robot end keeps the same position, moving direction and speed for dynamic grasping [19-22]. The control process is shown in Figure 19.

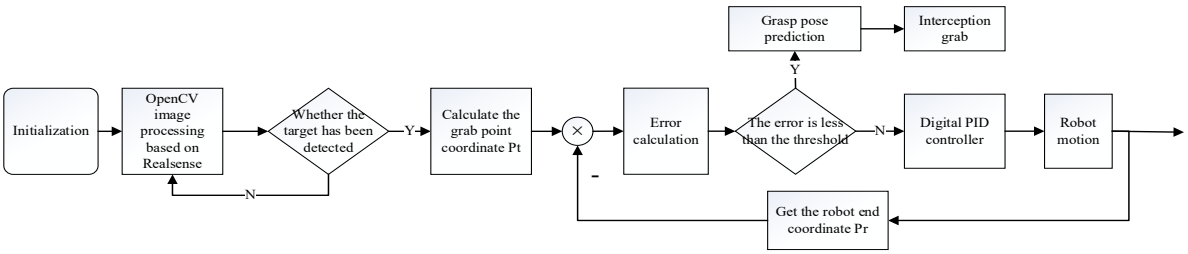


Figure 19. Automatic control principal block diagram

The dynamic target capture process is shown in Figure 20:

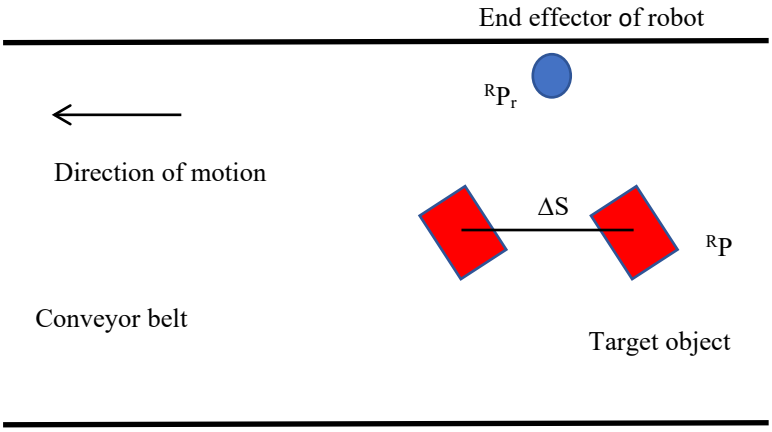


Figure 20. Conveyor belt tracking analysis

When the target enters the field of view of the camera, the first image acquisition and target recognition are carried out. The three-dimensional coordinate of the identified target in the base coordinate system is $^{R}\mathbf{P_0}$, the joint state of the robot arm at this moment is obtained, and the end position of the robot arm $^{R}\mathbf{P_{r0}}$ is obtained. The speed of the conveyor belt is **v**, and after time **t** , the displacement of the target object becomes $^{R}\mathbf{P_1}=^{R}\mathbf{P_0}+\mathbf{\Delta s}$, where $\mathbf{\Delta s=v*t}$ .The deviation between the two in the process of movement is expressed as shown in Equation (8):

$$\varepsilon(t)=^{R}P_0+\Delta s-^{R}P_{r0} \Rightarrow \varepsilon(t)=^{R}P_0+\mathrm{v}*t-^{R}P_{r0} \tag{8}$$

During the movement of the object, only translation transformation occurs, and the pose remains unchanged. The Angle between the conveyor belt and the X-axis is used to **θ**. The angle between the conveyor belt and the XOY plane is γ . Then the component error of the target in the three directions is expressed as shown in Equation (9):

$$\begin{cases} \varepsilon_{\mathrm{x}}(t)=x_{r0}-x_0+v*t*\sin\theta*\cos\gamma \\ \varepsilon_{\mathrm{y}}(t)=y_{r0}-y_0+v*t*\cos\theta*\cos\gamma \\ \varepsilon_{\mathrm{z}}(t)=z_{r0}-z_0+v*t*\sin\gamma \end{cases} \tag{9}$$

In this system, the belt coordinate system is parallel to the XOY plane. Therefore $\gamma=0$, the above equationcan be simplified as shown in Equation (10):

$$\begin{cases} \varepsilon_{\mathrm{x}}(t)=x_{r0}-x_0+v*t*\sin\theta \\ \varepsilon_{\mathrm{y}}(t)=y_{r0}-y_0+v*t*\cos\theta \\ \varepsilon_{\mathrm{z}}(t)=z_{r0}-z_0 \end{cases} \tag{10}$$

In the next cycle, according to the PID control principle, the position of the end of the robot can be obtained, $^{R}P_{n+1}$ can be expressed as shown in Equation (11) :

$$^{R}P_{n+1}=^{R}P_n+m(t) \tag{11}$$

It is necessary to use a discrete method to integrate the time, every sampling period **Δt**, and update $\mathbf{P_{n+1}}$. If the sampling period is short enough, equation (17) is regarded as continuous, then the adjustment quantity **m(t)** of the mechanical arm at time t is expressed as shown in Equation (12):

$$m(t)=K_p\left[\varepsilon(t)+\frac{1}{T_i}\Delta t*\varepsilon(t)+T_d\frac{\varepsilon(t)}{\Delta t}\right] \tag{12}$$

Where: m(t) is the adjustment quantity of the system at time t; $\mathbf{K_p}$ is the scale coefficient of the system; $\mathbf{T_i}$ and $\mathbf{T_d}$ are the integral and differential time constants of the system, respectively.

The components of the X, Y, and Z axes can be expressed as shown in Equation (13):

$$\begin{cases} m_x(t) = K_{xp}\left[\varepsilon_x(t) + \dfrac{1}{T_{xi}}\Delta t * \varepsilon_x(t) + T_{xd}\dfrac{\varepsilon_x(t)}{\Delta t}\right] \\ m_y(t) = K_{yp}\left[\varepsilon_y(t) + \dfrac{1}{T_{yi}}\Delta t * \varepsilon_y(t) + T_{yd}\dfrac{\varepsilon_y(t)}{\Delta t}\right] \\ m_z(t) = K_{zp}\left[\varepsilon_z(t) + \dfrac{1}{T_{zi}}\Delta t * \varepsilon_z(t) + T_{zd}\dfrac{\varepsilon_z(t)}{\Delta t}\right] \end{cases} \tag{13}$$

Finally, in the python control program of ROS, the PID algorithm is discretized, as shown in Equation (14):

$$\begin{cases} m_{xk} = K_{xp}\varepsilon_{xk} + K_{xi}\sum_{j=0}^{k}\varepsilon_{xk} + K_{xd}(\varepsilon_{xk} - \varepsilon_{x(k-1)}) \\ m_{yk} = K_{yp}\varepsilon_{yk} + K_{yi}\sum_{j=0}^{k}\varepsilon_{yk} + K_{yd}(\varepsilon_{yk} - \varepsilon_{y(k-1)}) \\ m_{zk} = K_{zp}\varepsilon_{zk} + K_{zi}\sum_{j=0}^{k}\varepsilon_{zk} + K_{zd}(\varepsilon_{zk} - \varepsilon_{z(k-1)}) \end{cases} \tag{14}$$

(1) The proportional coefficient Kp directly determines the strength of the control action, and increasing Kp can reduce the steady-state error of the system;

(2) On the basis of the proportional link, the integral link controlled by the integral time constant Ti can eliminate the steady-state error of the system, but the overshoot of the system will increase if the control is too strong, resulting in the deterioration of the stability of the system;

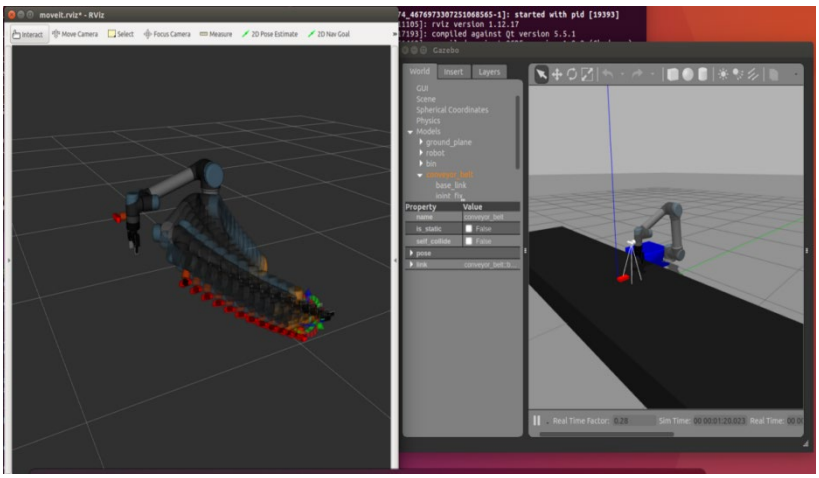
(3) Differential control can predict the deviation, and the larger the differential time constant Td, the faster the response speed and the larger the noise signal.

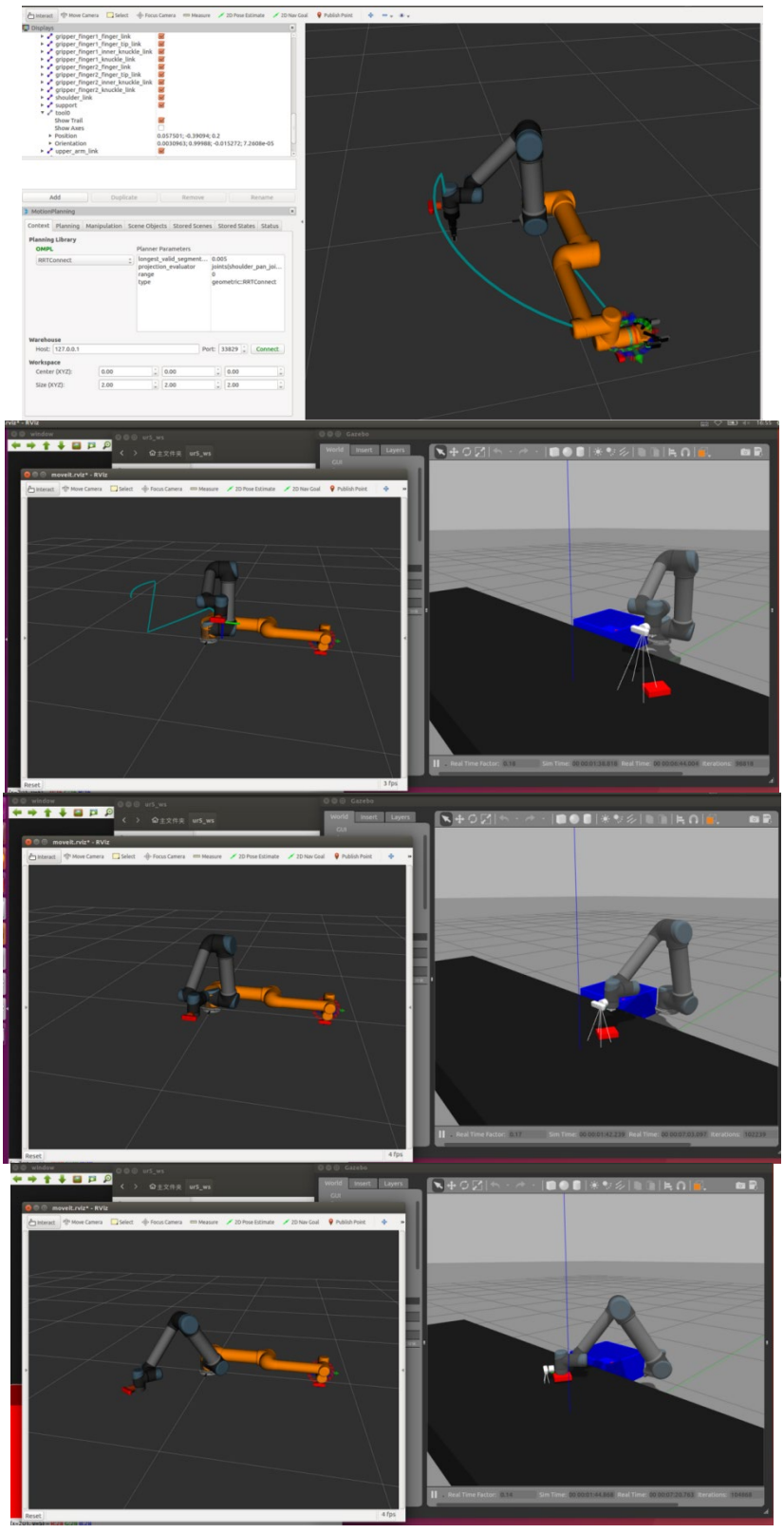
In the process of realizing dynamic tracking and grasping, when the motion state of the target changes, the distance between the robot arm and the target is constantly adjusted, and finally the direction and speed of the target are always consistent, and then the grasping action is carried out. In general, the PID dynamic tracking effect is good, and the trajectory meets the requirements.

3.0 RESULTS AND DISCUSSION

To validate the system's functionality, the target objects are sorted along the conveyor belt's trajectory. Vision-based detection identifies the targets, determining their poses based on spatial positions at the moment of capture. Subsequently, this information is relayed to the robot, enabling it to dynamically grasp the targets. Throughout the grasping operation, the robot's tracking coordinates are synchronized with the conveyor belt's speed and the object's three-dimensional coordinates captured during conveyor movement.

As depicted in Figure 21, the trajectory diagram illustrates the running path of the robot end effector's coordinate point. Through trajectory tracking tests, the system demonstrates its ability to stabilize movement effectively.





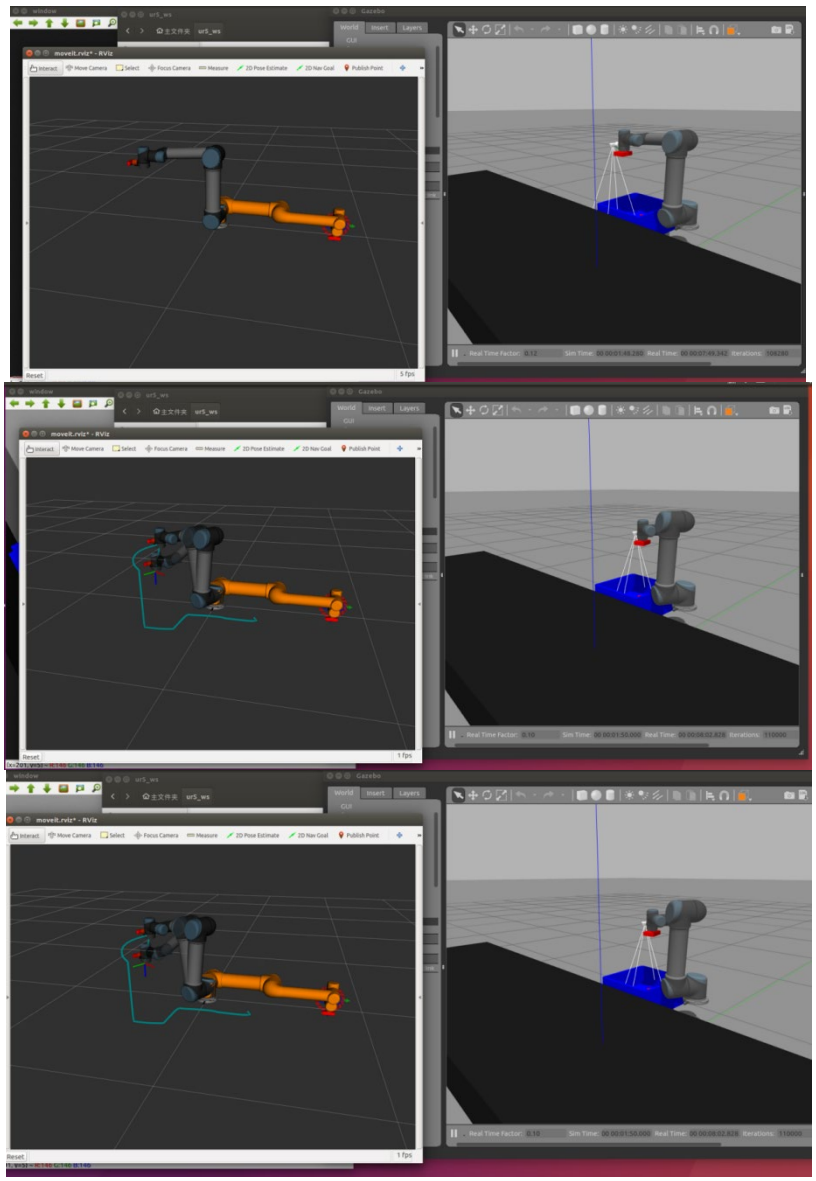


Figure 21. Conveyor belt tracking analysis

As depicted in Figure 22, the three images above display the camera's field of view. The red target's center coordinates are indicated post-recognition. The three images below showcase the grasping operation of the robotic arm. The robot arm's operational sequence unfolds as follows: initially, the robot arm moves to a predefined position and awaits the target's entry into the visual camera's field of view. Subsequently, the camera identifies the target and captures its positional deviation in the camera coordinate system (the positional deviation in x, y, and z coordinates between the target's center point and the camera's optical center), guiding the robot arm towards the target via visual feedback. As the conveyor belt advances, the robotic arm progressively closes in on the target, initiating the grasp once the pose deviation falls within the permissible range.

Run the system program to verify the system functions. The implementation process of dynamic target capture is shown in Figure 22. It can be seen that with the help of the experimental platform, the dynamic target machine vision recognition and the grasping system of the robot arm can complete a cyclic process from tracking, approaching, grasping, placing to grasping all the dynamic targets on the conveyor belt. Realize the recognition and capture of dynamic objects.

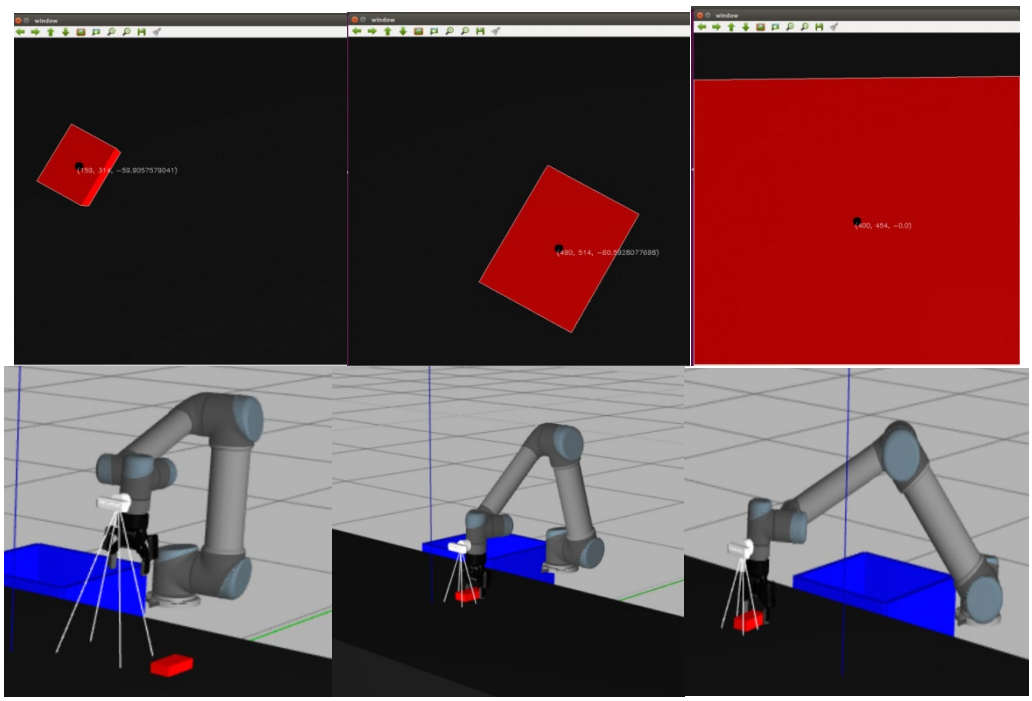


Figure 22. Gazebo dynamic capture work

During the dynamic grasping process of the manipulator, alterations in the target object and the manipulator exclusively manifest in the XOY plane, with no orientation changes relative to the initial state. Based on the aforementioned formula, the system generates the corresponding response for the dynamic grasping process. Throughout this process, the conveyor belt maintains a constant speed, resulting in the following system response error.

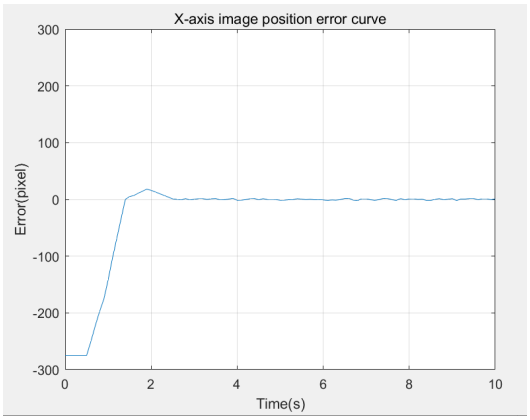


Figure 23. X-axis image position error

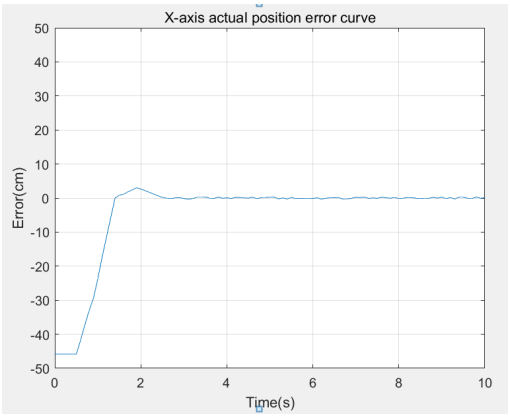


Figure 24. X-axis actual position error

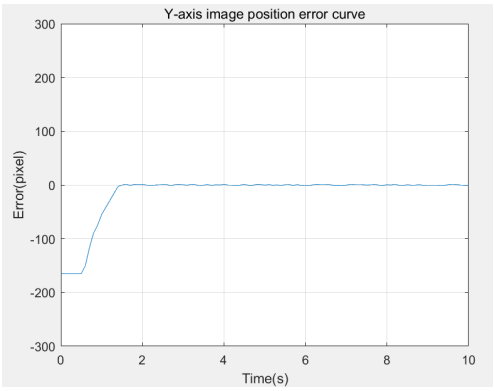


Figure 25. X-axis image position error

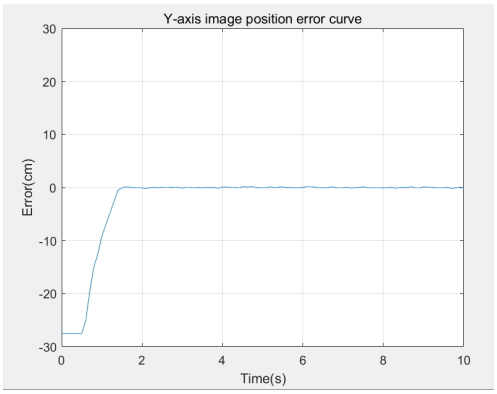


Figure 26. X-axis actual position error

By adjusting the control parameters, it can be seen from the above figure that in the early stage of dynamic target tracking, the displacement error between the robot and the target is large. As time goes on, the distance between the two gradually decreases and gradually tends to zero, realizing the unity of their position and then completing the grasp.

In order to verify the function of the system, different workpieces (including 100 target workpieces) are randomly distributed on the conveyor belt, and the conveyor belt is set to run at four different speeds of 10mm/s, 20mm/s, 30mm/s and 40mm/s, run the system program, measure the time used to grasp all dynamic targets, and check the number of missing grasp target workpieces of the system. Verify the validity of system function.

Table 3. Dynamic target capture effect analysis

Speed(mm/s)	Total amount	Successful number	Grasp failure rate
10	100	100	0%
20	100	97	3%
30	100	93	7%
40	100	90	10%

As can be seen from the dynamic grasp effect, when the speed of the conveyor belt is slow, the grasp time is longer, but the success rate is high, and the missed detection is few. With the continuous improvement of the speed of the conveyor belt, the dynamic grasp efficiency has been greatly improved, but with the rapidity of the grasp, the number of missed grasp of the target object is also increasing, but the system as a whole, the success rate of dynamic grasp is maintained at more than 90%. Meet system reliability.

4.0 CONFLICT OF INETEREST

Thea author declare no conflicts of interest.

5.0 AUTHORS CONTRIBUTION

Zhang XiaoYang (Formal analysis; Methodology; Validation; Data analysis and Writing - original draft)

Muhammad Azmi Ayub (Supervision; Industrial robot and machine vision methodology; Resources)

Fazlina Ahmat Ruslan (Machine vision algorithm; Paper review and proofreading)

Sukarnur Che Abdullah (Industrial robot trajectory planning algorithm)

Shuzlina Abdul-Rahman (ROS system industrial robot simulation design)

6.0 ACKNOWLEDGEMENT

This research received grants from Hebei Institute of Mechanical and Electrical Technology in China. Furthermore, this research was also supported by the UiTM collaborative research laboratory.

7.0 REFERENCES

[1] Wang T M, Tao Y. Status Quo and Industrialization Development Strategy of Industrial Robot Technology in China. [J]. Journal of Mechanical Engineering. 2014;50(09):1-13.

[2] Li X F, Zhuang J P, Gao Y, Wu K L. A dynamic tracking algorithm for robot conveyor belt. Machine Tools and Hydraulics. 2021;49 (17) :65-7+73.

[3] Tian C. Research on the status quo and development trend of industrial robots. [J]. China Management Information Technology, 2019.22(20):156-7.

[4] Xu W H. Research on intelligent recognition of moving objects in dynamic environment based on machine vision. Laser Journal. 2022;4301):29-32.

[5] Geng L M, Wang D, Yang W. Research on machine vision in intelligent sorting and recognition of industrial robots. Electronic Production. 2018(20):32-3.

[6] Ashraf MA, Kondo N, Shiigi T. Use of Machine Vision to Sort Tomato Seedlings for Grafting Robot. Engineering in Agriculture, Environment and Food. 2011;4(4).

[7] TOSEPH L. Mastering ROS for robotics programming: design,build and simulate complex robots using robot operating system and master its out-of-the-box functionalities [M]. 2nd ed.Birmingham: Packet Publishing,2018.

[8] Wang Y Y, Wang L M, Guan J W, Wang F X. Indoor transportation automated guided vehicle system based on robot operating system and its design [J].Science Technology and Engineering, 2020,20(19):7742-9.

[9] Gong Z L, Gu Y H, Zhu T T, Ren B. A method for setting costmap adaptive inflation radius based on robot operating system [J]. Science Technology and Engineering, 2021;21(09):3662-8.

[10] Han Y X, Zhang Z S, Dai M. Monocular visual measurement method for target ranging [J]. Optics and Precision Engineering,2011;19(05):1110-7.

[11] Golnabi H, Asadpour A. Design and application of industrial machine vision systems [J]. Robotics and Computer Integrated Manufacturing. 2007;23(6):630-637.

[12] Pang C T. Research on Robot Tracking and Grasping Technology based on Visual servo [D]: Jiangsu University; 2021.

[13] Yang S B, Long Y H, Xiang Z Y, Yao J C. Research on Binocular vision stereo matching based on SURF algorithm [J]. Journal of Hunan University of Technology, 2019;33(03):75-80.

[14] Wang J H. Research on close-range 3D measurement technology based on laser fringe [D]. Harbin Institute of Technology; 2019.

[15] Ma S C. Research on Robot Grasping System Based on Visual Inspection [D]: Tianjin University of Technology; 2020.

[16] Cheng Y L. Robot Hand-eye Calibration and Object Localization for Industrial Applications [D].2016.

[17] Kuang J H, He Y B, Chen R L. Design of OpenCV based target grasping for conveyor belt. Manufactured and upgraded today. 2023(11):93-5.

[18] Zhang L, Zhou J G, He W. Implementation of PID Control Algorithm in Video tracking System [J]. Industrial Control Computer,2007,20(7):15-16.

[19] Deng M X, Liu G F, Zhang G Y. Dynamic Tracking of Conveyor Belt Based on Delta Parallel Robot [J]. Mechanical Engineering and Automation, 2015(01):153-154.

[20] Wang Z, Dai J F, Qian Z Y, Shou K R. Research on Robot target Tracking and Grasping Strategy for Conveyor Belt Operating System [J]. Computer Measurement and Control, 2016,24(11): 85-90.

[21] He Z Q. Design of Industrial Robot Sorting System based on Machine Vision [D]. Harbin Institute of Technology; 2016.

[22] 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.