Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Research on Autonomous Grasping Technology Based on Vision*

Yilin Zhao[1)], Pengcheng Xue[2)], Wenhui Cai[1)], Junsheng Wu[2)], Tian Ai[2)] and Weijun Zhang[#2)]

[1)] *Kunming Power Supply Bureau*
Yunnan Power Grid Co.,Ltd.
Kunming, China
85989931@qq.com

[2)]*School of Mechanical Engineering*
Shanghai Jiao Tong University
Shanghai, China
xuepch@sjtu.edu.cn

Abstract—The grasping operation is limited by visual observational conditions, operator level, etc., in the process of robot teleoperation. In response to this problem, this paper proposes a method of autonomous grasping based on vision. A depth camera is carried by the robot to obtain the color and depth information of the environment. The operator selects the target with a box on the monitor to determine the approximate range of the target. Combining the target box, depth information and color information, the image foreground is extracted to obtain the plane contour and depth information of the target. Then, the pose data of the target are calculated, and the grasping position is estimated and converted into a position in the robot coordinate system. After the grasping position is determined, the grasping trajectory of the manipulator is planned and optimized, combined with the environmental obstacle information. Finally, the manipulator is controlled to complete the grasping process. Experimental results show that the autonomous trajectory planning of the robot is continuous and controllable and that the grasping process can be completed efficiently with high accuracy in the recognition and positioning processes.

Keywords—Autonomous Grasping; Visual Guidance; Trajectory Planning

I. Introduction

Explosive ordnance disposal (EOD) robots have become an excellent choice for handing suspicious explosive objects. Its application can reduce the danger faced by explosive disposal personnel and prevent unnecessary casualties. When used in unknown environments, EOD robots face many challenges. Due to the flatness of the image the robot works with, it is difficult to judge the grasping position of the target. Furthermore, as a result of the unpredictability in the operating process, operators need to make timely judgments according to the environment, which requires operators to undergo numerous training sessions for this problem. Additionally, the grasping speed cannot be guaranteed because of the inconsistency of human operation. Therefore, it is important to propose a method that can realize autonomous grasping.

Many studies on autonomous grasping have been conducted using different methods. One common approach is a visual guidance algorithm based on RGB-D point-cloud processing[1,2]. The color image and depth information are combined to segment a point cloud, and then the results are screened and evaluated to calculate the grasping positions for 3d objects. Point cloud segmentation is illustrated in Fig. 1. RanSaC, 3D Hough Transform and table-top are commonly used algorithms for point cloud segmentation. The table-top algorithm first performs plane segmentation, removes points that do not meet the requirements according to the normal vector and neighborhood information of points, and performs clustering on the remaining points to obtain the center of the grasping target. The object detection methods in the existing research require a large amount of calculation, and the object features must be modeled in advance.

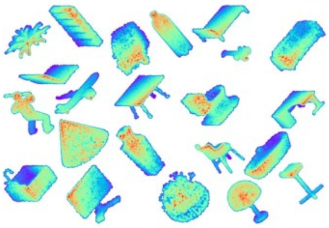


Fig. 1. Point cloud segmentation

This paper uses an EOD robot as the research object to study the visual grasping algorithm of a manipulator. First, we present a method to identify and segment the target and calculate the position of the grasping point by means of a depth camera and human-computer interaction. Then, a method to plan and optimize the grasping trajectory of the manipulator is presented. Finally, we verify the methods by experiments.

II. Target Localization

After the robot moves to the designated position, it needs to locate and grasp the target in human-computer interaction mode. First, a box is drawn on the outside of the target, and the target contour is obtained after image segmentation. Then, the image of the target is binarized, and the position of its centroid is calculated. The position of the centroid is the final grasping position of the manipulator. According to the requirements and characteristics of image processing, the image processing algorithm designed in this paper is shown in Fig. 2.

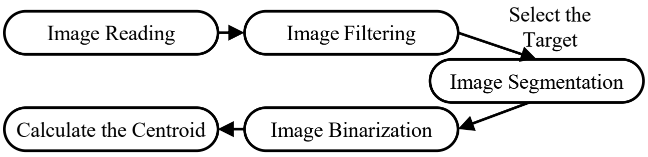


Fig. 2. Process of image processing

* This work is partially supported by China Southern Power Grid (YNKJXM20170208)
# Corresponding author, zhangweijun@sjtu.edu.cn

A. Image Segmentation

To better obtain the location of the object, the operator needs to draw a box on the outside of the target, and then the target contour will be extracted through the image segmentation algorithm. Image segmentation technology divides images into several nonintersecting regions according to the characteristics of the image itself, such as color and texture, with low similarity between regions and high similarity within regions. This paper adopts the GrabCut[3,4] algorithm for image segmentation.

1) Image segmentation based on GrabCut

The GrabCut algorithm is based on the color information of the image, working more effectively when there is a contrast between the color of the target and the background. The GrabCut algorithm is essentially a segmentation method based on graph theory; that is, the image is converted into a graph structure to achieve the requirements of image matting. The graph structure is composed of vertices and edges, that is, the picture can be represented as a graph G=(V,E). The GrabCut algorithm converts all pixels in the image into vertices in the graph structure and adds two vertices on this basis, namely, a source point S and a sink point T. The vertex connected with source point S after final segmentation is the target, and the vertex connected with sink point T is the background. There are also two kinds of edge E. The first is n-links, which connect pixels to pixels in the image, and the other is t-links, which connect the source point or the sink point to the pixels. The network diagram created from the original diagram is shown in Fig. 3.

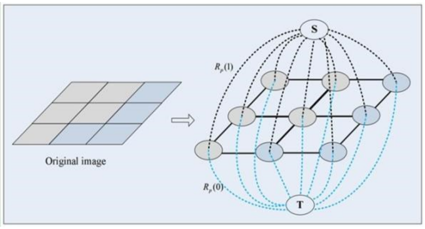


Fig. 3. GrabCut algorithm model

Each edge in the graph structure corresponds to a cost. Different segmentation methods will yield the sum of costs of different sizes (losses). The smaller the loss, the better the corresponding segmentation effect, that is, the more reasonable the segmentation of the target and the background. By constructing the graph structure, the original image segmentation problem is transformed into a maximum flow/minimum cut problem in the network flow, so a maximum flow/minimum cut solution algorithm can be used to find the optimal segmentation position. The GrabCut algorithm uses the Gaussian mixture model (GMM) to represent the target and background in the color image, assuming that the color space (R, G, B) conforms to a Gaussian distribution [5]. The Gaussian mixture model needs specific parameters to be determined, and different parameters will correspond to the losses caused by different segmentation. Parameters are calculated in an iterative manner and are updated along the direction of loss reduction.

Assuming that the whole image conforms to a Gaussian mixture model that consists of K Gaussian components, the whole image can be represented by a vector, namely $\left(k_1, k_2 ..., k_n\right)$, where the i th component k_i means that the i th pixel belongs to the Gaussian component k_i. The length of the vector is the number of pixel points in the image, and any pixel in the image can only correspond to one Gaussian component. According to the established mixed Gaussian model, the loss of the defining image segmentation is shown as follows.

$$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z) \tag{1}$$

$U(\alpha, k, \theta, z)$ is the regional item, which describes the region attributes. If a pixel belongs to the target, it is expected that its corresponding Gaussian distribution has a large weight, that is, it will lose less if classified as the target, and vice versa. The regional terms are defined in (2), (3), and (4).

$$U(\alpha, k, \theta, z) = \sum D(\alpha_n, k_n, \theta_n, z_n) \tag{2}$$

$$\begin{aligned} D(\alpha_n, k_n, \theta_n, z_n) &= -\log \pi(\alpha_n, k_n) + \frac{\log \det \Sigma(\alpha_n, k_n)}{2} \\ &+ \frac{\left[z_n - \mu(\alpha_n, k_n)\right]^T \Sigma(\alpha_n, k_n)^{-1} \left[z_n - \mu(\alpha_n, k_n)\right]}{2} \end{aligned} \tag{3}$$

$$\theta = \left\{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, k = 1 \ldots K\right\} \tag{4}$$

The logarithm in (3) is taken to ensure that the more similar the pixel is to a Gaussian distribution in the regional term, the lower the cost of dividing it into the region. In (1), $V(\alpha, z)$, the boundary term, describes a border attribute that evaluates the similarity between pixels. If the gray values of the two adjacent pixels are very close, the probability that these two pixels belong to the target or the background is very high. If the gray values of these two pixels are very different, then these two pixels are likely to be the boundary between the target and the background. Therefore, when two pixels have high similarity but are divided, a large loss will be generated, and vice versa. The boundary term is defined in (5):

$$V(\alpha, z) = \Upsilon \sum_{(m,n) \in C} \left[\alpha_n \neq \alpha_m\right] exp - \beta \left\| z_m - z_n \right\|^2 \tag{5}$$

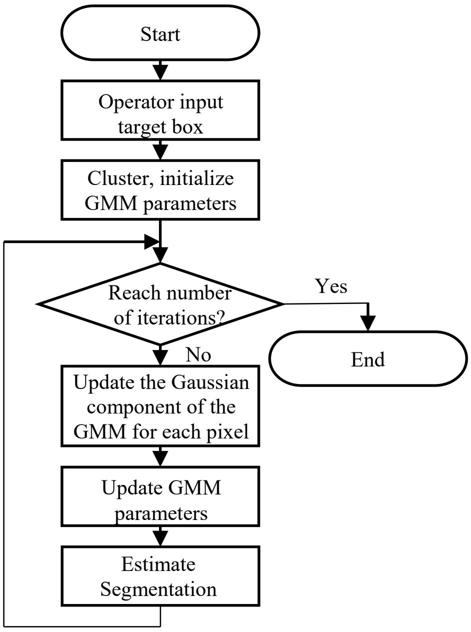
$\left\| z_m - z_n \right\|^2$ is the Euclidean distance, used to describe the similarity of the ray value between the pixels. Constant Υ is usually set to 50. β is the attenuation factor, which is adjusted according to the contrast of the image. When the contrast of the image is low, β is large, that is, the average value of the boundary term is reduced to reduce its contribution to the final loss. When contrast is high, β is small.

There are three parameters in the Gaussian mixture model that need to be determined: the weight of the Gaussian component π, the mean vector μ and the covariance matrix Σ. A color image is composed of three RGB colors, so the dimension of the mean vector μ is 3 and the covariance matrix Σ has three columns and three lines. A total of 6 parameters need to be determined for the target and background Gaussian mixture models. After the parameters are determined, a pixel value is substituted into the two Gaussian mixture models, and the corresponding similarity can be calculated to determine whether the pixel belongs to the target or the background; the corresponding confidence can be described with the similarity.

2) *Solving the GrabCut Algorithm*

According to the above established model, the GrabCut algorithm attempts to continuously update the parameters of the mixed Gaussian model to reduce the loss. An iterative method is used to solve the problem, and the process is shown in Fig. 4. The target box inputted by the operator is taken as the initial reference. The region outside the target box belongs to the background, and its corresponding parameter $\alpha_n = 0$, while the region inside the target box belongs to the target, and its corresponding parameter $\alpha_n = 1$. According to the initial target region and background region, the corresponding mixed Gaussian model parameters are calculated. First, the K-means clustering algorithm is used to cluster the pixels in the region into K classes, corresponding to K components in the mixed Gaussian model. The corresponding weight is the ratio of the number of pixel points in the region to the total number of pixels. The mean value and covariance of the model are calculated according to the gray values of the three RGB channels. Then, the Gaussian component of the corresponding Gaussian mixture model is calculated for each pixel. If the i th pixel belongs to the background, the RGB gray value of the pixel is substituted into the Gaussian mixture model of the background to calculate and select the component with the largest value among the obtained components:

$$k_{\mathrm{i}} = \arg\min_{k_i} D_i\left(\alpha_i, k_i, \theta_i, z_i\right) \tag{6}$$



Start
Operator input target box
Cluster, initialize GMM parameters
Reach number of iterations?
Yes
No
End
Update the Gaussian component of the GMM for each pixel
Update GMM parameters
Estimate Segmentation

Fig. 4. Process of GrabCut algorithm

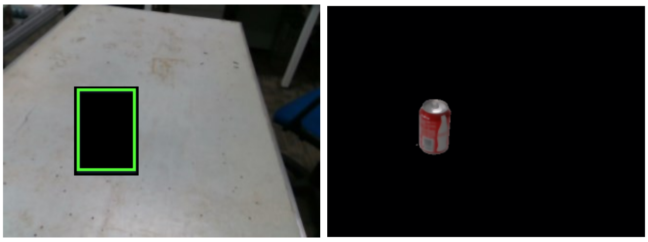
According to the calculated Gaussian component, the mixed Gaussian model is updated:

$$\theta = \arg\min_{\theta} U\left(\alpha, k, \theta, z\right) \tag{7}$$

After each update of the mixed Gaussian model parameters, a segmentation estimation should be made; that is, according to the loss calculated by (2), the structure of a graph is constructed, where the weight of the edge is the calculation result of this equation, and then segmentation is performed by the maximum flow/minimum cut algorithm:

$$\min_{\{\alpha_n : n \in T_U\}} \min_{k} E\left(\alpha, k, \theta, z\right) \tag{8}$$

On the filtered image, the target is selected with a box to segment the color image. The segmentation results are shown in Fig. 5.



(a) Box around the target

(b) Target extracted

Fig. 5. Image segmentation

B. *Calculating Grasping Position*

After image segmentation algorithm processing, the background region is all set to black, with the target color unchanged. Since the target image is in color, the entire image after segmentation must be set to grayscale and then binarized. Then, the position of the centroid of the target is calculated by the image moment.

Image binarization converts an image to a pure black and white image with all pixel values set to either 0 or 255. The process of binarization compares the gray value of each pixel with a threshold value to determine which category the point belongs to. The final effect of binarization has a great relationship with the selection of thresholds, and there are many methods for threshold selection. This paper adopts the global binarization algorithm based on the image histogram, which first calculates the maximum and minimum gray values of the image, denoted as g_{max} and g_{min}, respectively, and initializes the threshold T_0 according to (9).

$$T_0 = \frac{g_{max} + g_{min}}{2} \tag{9}$$

Then, the image is segmented into two regions according to the threshold value, and the average gray value H_b and H_f of the two regions are calculated according to (10)

$$\begin{cases} H_b = \dfrac{\sum_{g=g_i}^{T_0} g \times h\left(g\right)}{\sum_{g=g_i}^{T_0} h\left(g\right)} \\ H_f = \dfrac{\sum_{g=T_0+1}^{g_u} g \times h\left(g\right)}{\sum_{g=T_0+1}^{g_u} g \times h\left(g\right)} \end{cases} \tag{10}$$

According to the segmented region, the threshold is updated:

$$T_k = \frac{H_b + H_f}{2} \tag{11}$$

The process of segmentation and threshold updating is repeated until the threshold no longer changes. The algorithm flow is shown in Fig. 6. The global binarization algorithm based on the image histogram does not need to set the threshold in advance and can automatically calculate the appropriate threshold for binarization for different images.

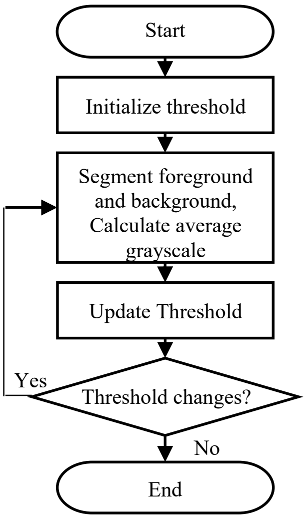


Fig. 6. Process of binarizing image

The binarization effect on the target segmented by GrabCut is shown in Fig. 7.

Fig. 7. Image binarization result

After binarization, the target and background are divided into black and white colors. For the target, it is necessary to determine a suitable position for grasping. Here, the centroid of the target is selected for grasping because grasping at the centroid can ensure that the target is as balanced as possible when grasped. Assuming that the mass of the target is uniformly distributed, the position of the centroid is obtained by solving the image moment. The mathematical formula of moment is shown in (12):

$$M_{pq} = \int_{a_1}^{a_2}\int_{b_1}^{b_2} x^p y^q f(x,y)\,dxdy \tag{12}$$

Low order moments such as M_{00}, M_{01} and M_{10} can be used to calculate the centroid. For an image with m rows and n columns of pixels, the target region is white (a gray value of 255) and the background is black (a gray value of 0), so the gray value of the background region has no influence on the calculation of the centroid. Since the image is a discrete region, the integral of (12) should be converted into a sum. A rectangular coordinate system is established with the pixel at the lower left corner of the image as the origin. The X-axis runs along the bottom of the image to the right, and the Y-axis runs along the left side of the image to the up. The gray value of point (m,n) is $F(m,n)$, and the centroid coordinate is (m_c, n_c), where

$$\begin{cases} m_c = \dfrac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} mF(m,n)}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(m,n)} \\ n_c = \dfrac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} nF(m,n)}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(m,n)} \end{cases} \tag{13}$$

According to the centroid calculated by the image moment and to the depth information, the calculated grasping point is shown in Fig. 8.

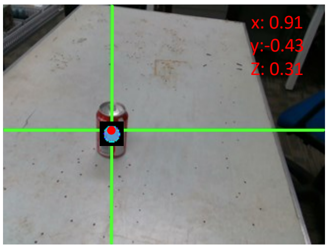


Fig. 8. Image centroid calculation result

III. AUTONOMOUS GRASPING

By properly positioning the target, the grasping point E (p_x, p_y, p_z) is obtained. Suppose the position of the manipulator before grasping is S. If the manipulator is directly moved from S to the grasping point E, it will collide with the target in some cases. Therefore, this paper performs two-stage trajectory planning by adding an intermediate point P as shown in Fig. 9. The intermediate point P is located at a distance d above the centroid of the target (the z-axis direction in the coordinate system at the base of the manipulator), where d is the physical height of the target box input by the operator: that is, the coordinates of point P are $(p_x, p_y, p_z + d)$. The trajectory planning from the starting point S to the intermediate point P is carried out in the joint space. The motion of this segment only needs to ensure that the final motion reaches point P without any other constraint. The trajectory planning from the intermediate point P to grasping point E is carried out in a Cartesian coordinate system, and it is necessary to ensure that the movement of the end effector is vertical.

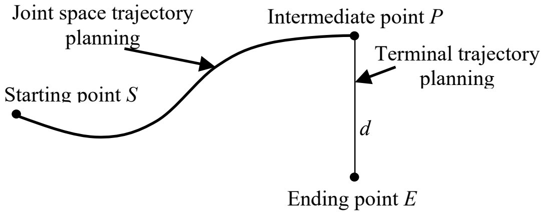


Fig. 9. Two-stage grasping trajectory planning

A. From Starting Point to Intermediate Point

First, the angle corresponding to each joint is inversely solved according to the coordinates of the intermediate point *P*, and then each joint is taken as a research object to plan from the starting point to the intermediate point. Suppose that joint i is at an angle of θ_{is} at the starting point, the reverse-solved joint i is at an angle of θ_{iP} at the intermediate point *P*, and cubic polynomials are used to plan from θ_{is} to θ_{iP} with a period of T. Then, the angle of the joint at any time is shown in (14):

$$q(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \tag{14}$$

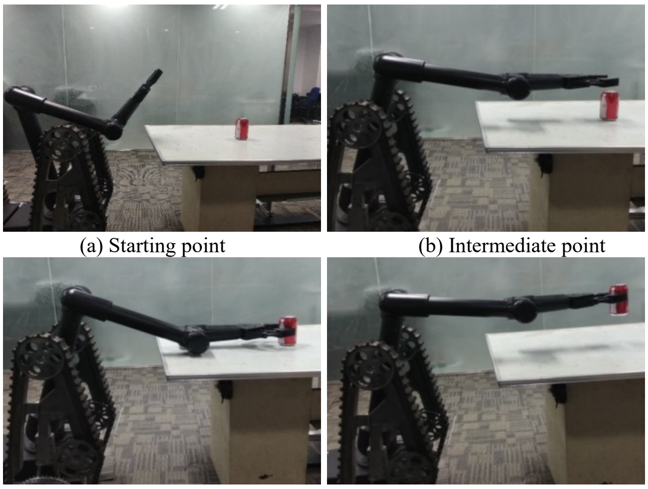
$q(t)$ represents the joint angle at time t. The four parameters in (14) need to be determined, so four equations need to be listed for the solution. Since the initial angle is known, the angle at the intermediate point is also known through the inverse solution of (14), and the velocity at the starting and ending moments is zero, the four equations obtained are shown in (15):

$$\begin{cases} a_3 = \theta_{is} \\ a_0 T^3 + a_1 T^2 + a_2 T + a_3 = \theta_{ip} \\ a_2 = 0 \\ 3a_0 T^2 + 2a_1 T + a_2 = 0 \end{cases} \tag{15}$$

According to (15), the position of each joint at each moment can be solved and distributed to the joint. The motion time of each joint is the same T, that is, all the joints start and stop at the same time.

B. From Intermediate Point to Grasping Point

The trajectory planning from the intermediate point to the grasping point is carried out at the end effector to ensure that the end of the end effector moves in a straight line. First, interpolation is carried out from intermediate point P to grasping point *E*. The number of interpolation points is N. The N points are inversely solved according to the inverse kinematics of the manipulator to obtain each joint angle. The straight motion of the end effector can be achieved by distributing these joint angles to the joints successively. The grasping process is shown in Fig. 10.



(a) Starting point (b) Intermediate point

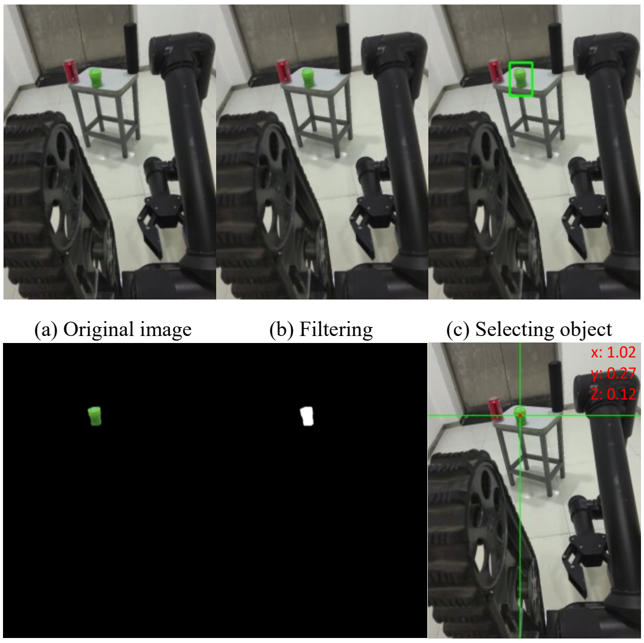
(c) Grasping point (d) Grasping object

Fig. 10. Process of object grasping

IV. Experiments And Analysis

A. Calculating the grasping position

First, the depth camera is used to read the image, as shown in Fig. 11 (a). Bilateral filtering is performed on the original image to remove noise, as shown in Fig. 11 (b). The target is then selected manually, as shown in Fig. 11 (c). Next, the GrabCut algorithm is used to extract the outline of the target, as shown in Fig. 11 (d). Then, the image is binarized, as shown in Fig. 11 (e). Finally, the centroid is calculated by the image moment, as shown in Fig. 11 (f).



(a) Original image (b) Filtering (c) Selecting object

(d) Image segmentation (e) Binarization (f) Calculating centroid

Fig. 11. Target extraction

In Fig. 11, the green bottle is selected with a box, and the grasping center is calculated. Fig. 11 (f) shows that grasping is feasible at the calculated centroid (the position of the red dot). After camera calibration and transformation, the obtained position of the grasping center relative to the coordinate system of the base of the manipulator is $(1.02, 0.27, 0.12)$.

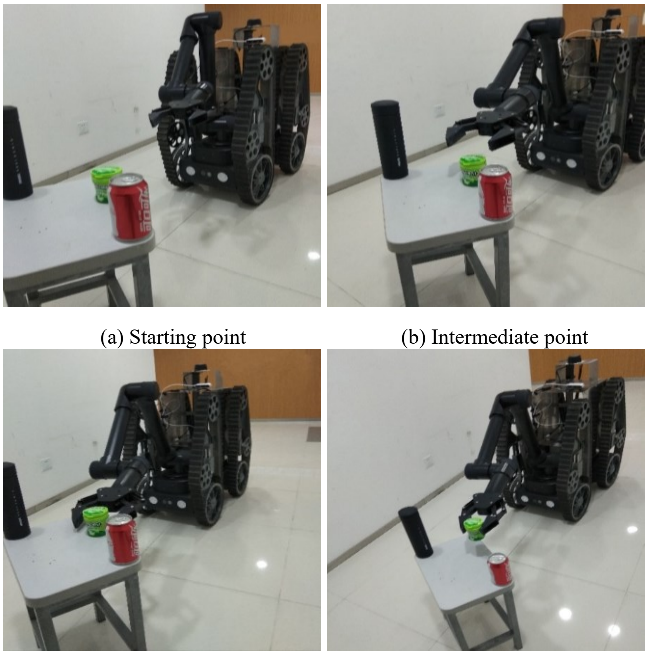
Under the spatial coordinates of the base of the manipulator, the position of the target is constantly changed. The deviations between the calculated centroid and the actual centroid are shown in table 1, and the errors are all less than 8 mm, meeting the requirements of grasping.

TABLE I. Centroid Calculation Deviation Table

i	x deviation(mm)	y deviation(mm)	z deviation(mm)
1	3.1	2.9	1.7
2	2.5	5.1	2.0
3	6.8	1.3	3.8
4	3.4	4.3	1.1
5	3.2	3.4	2.0
6	1.1	0.7	0.1
7	2.9	3.2	2.2
8	5.3	4.8	7.1

B. *Trajectory Planning of the End Effector*

After obtaining the coordinates of the centroid of the target, the joint angles are solved by inverse kinematics of the manipulator. The position of the manipulator before grasping is the initial position, as shown in Fig. 12 (a). The waist joint angle of the manipulator is 0°, the big arm joint angle is -83.4°, the middle arm joint angle is 110.6°, the small arm joint angle is -30.0°, and the wrist joint angle is 0°. The corresponding joint angle of the target position is calculated according to the current joint angle and the target point coordinates. The calculated result is a waist joint angle of 15.0°, a big arm joint angle of -43.4°, a middle arm joint angle of 75.6°, a small arm joint angle of -30.0°, and a wrist joint angle of 0°. The coordinate of the intermediate point is set as $(1.02, 0.27, 0.22)$, that is, the position 10 cm above the grasping point, as shown in Fig. 12 (b). At the intermediate point, the waist joint angle is 14.8°, the big arm joint angle is -49.5°, the middle arm joint angle is 77.3°, the small arm joint angle is -30.0°, and the wrist joint angle is 0°. After reaching the intermediate point and pausing for 1 s, the end effector moves to the grasping position and lifts, as shown in Fig. 12 (c) and Fig. 12 (d).

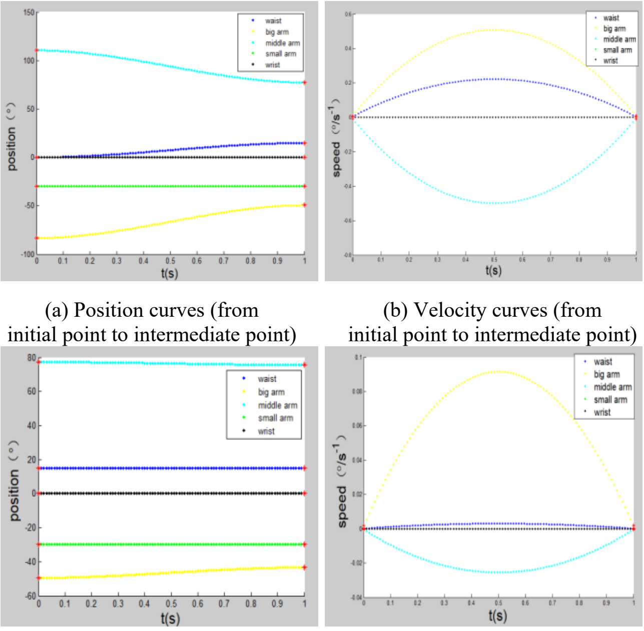


(a) Starting point (b) Intermediate point

(c) Grasping position (d) Lifting

Fig. 12. Manipulator end effector trajectory planning experiment

The manipulator grasping trajectory planning is divided into two segments of motion. In the first segment, the manipulator moves from the initial position to the intermediate point. In the second segment, the manipulator moves from the intermediate point to the grasping position. During the movement of the manipulator, the angles and angular velocities of the waist joint, the big arm joint, the middle arm joint, the small arm joint and the wrist joint are recorded, and the angles and angular velocity curves are shown in Fig. 13; during the planning of the end effector of the manipulator, it can be seen that the position curve and velocity curve are both continuous and smooth, and the whole manipulator is not impacted during the movement.



(a) Position curves (from initial point to intermediate point)

(b) Velocity curves (from initial point to intermediate point)

(c) Position curves (from intermediate point to grasping point)

(d) Velocity curves (from intermediate point to grasping point)

Fig. 13. Manipulator joint motion status

Conclusion

In this paper, the grasping position of the target is obtained through image segmentation. During the grasping process, the motion of each joint of the robot arm is continuous and is not impacted. This method performs interactive intelligent teleoperation; however, it relies only on color information for image segmentation. When the color information of the target and the background object are similar, the segmentation effect will be reduced. Future research will combine color information and depth information to provide different weights and adapt to different scenes.

References

[1] M. Saxena A, Driemeyer J, Kearns J, et al. "Learning to Grasp Novel Objects Using Vision"// Experimental Robotics. Springer Berlin Heidelberg, 2008: 33-42.

[2] C. Hsiao E, Collet A, Hebert M. "Making specific features less discriminative to improve point-based 3D object recognition". Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition, 2010: 2653-2660.

[3] J. Shaobing Yang, Leiming Li, et al. "Adaptive Background Image Segmentation Algorithm Based on GrabCut". Computer system and application. 2017, 26(2): 174-178.

[4] D. Yuning Zhang. "Color Image Segmentation Based on Contrast and Grab Cut ". Hebei University, 2018.

[5] C. YY Boykov, MP Jolly. Interactive Graph Cuts for Optimal Boundary& Region Segmentation of Objects in N-D Images[C]//English IEEE International Conference on Computer Vision. New York: IEEE computer Society Press, 2001: 105-112.

[6] C. Heller, Jan, Henrion, Didier, Pajdla, Tomas Hand-eye and robot-world calibration by global polynomial optimization//IEEE International Conference on Robotics & Automation. 2014: 3157-3164.