A GUI Software for Automatic Assembly Based on Machine Vision

Hao-dong Chen, Yi-fan Wang
School of Mechanical Engineering
Hefei University of Technology
Hefei, China
e-mail: haodong-chen@mail.hfut.edu.cn

Zheng Guo
School of Mechanical Engineering
Shanghai Jiao Tong University
Shanghai, China
e-mail: guo_zheng@sjtu.edu.cn

Wen-xiu Chen, Ping Zhao*
School of Mechanical Engineering
Hefei University of Technology
Hefei, China
e-mail: ping.zhao@hfut.edu.cn

***Abstract*—In this paper, a graphical user interface (GUI) is devised to perform efficient and accurate object recognition and real-time path planning of parallel manipulator in the packaging and assembly industry production. The selection of reference objects, workspace, and other parameters are involved in this GUI. Target recognition, path planning and simulation of assembly can also be carried out. A short description of the general algorithm underlying the GUI is given. And the full functionality of the algorithm is captured in the high level of abstraction in this GUI. The simulation results show the validity and feasibility of the GUI and the algorithm.**

Keywords-GUI; assembly; object recognition; image matching; path planning

I. INTRODUCTION

Figure 1. IRB 340 FlexPicker delta robot.

These days, in order to reduce the production cost and satisfy the demand of customers, parallel robots have a wide range of applications in the packaging and assembly industry production. In 1942, a parallel mechanism was proposed by Willard L and patented later [1]. In 1965, a six degree-of-freedom parallel platform was proposed by Stewart in his paper [2]. In the early 80's, inspired by parallelograms, Reymond Clavel came up with the brilliant idea of Delta robot with three translational and one rotational degree-of-freedom [3]. And in 1999, a new model of Delta robot with four-degree-of-freedom, named IRB 340 FlexPicker (Fig.1), was launched by ABB Flexible Automation [4]. At the same time, a modified version, NUWAR, was built by University of Western Australia [5]. The axes of its rotary actuators were non-coplanar arranged.

In the application of parallel robots, to realize the given accuracy and efficiency, computer vision is adopt to realize the given accuracy and efficiency. Visual detection is one of the most challenging issues in computer vision. In the early time, the initial work mainly concentrated on the binary image analysis and identification [6]. In 1963, by extracting three-dimensional structures of polyhedron, Roberts Cross detector was proposed to describe the spatial relationship between the shapes of the objects. However, it was vulnerable to image noise [7]. In 1968, Sobel algorithm was proposed, which had a 3x3 isotropic gradient operator for image smooth, but with low accuracy of edge location [8]. In the mid-1986s, a multilevel edge detection algorithm, Canny algorithm, was developed by John F. Canny [9-10]. In 2009, a new shape signature for Fourier Descriptors is proposed by Elghazal [11]. In this paper an approach based on Fourier Descriptors (FDs) and Canny algorithm is applied to accomplish object recognition. After that, assembly of special pose and location can be realized by image matching approach. Image matching approach is used to decide which object model positions (including rotation and scaling) are good matches to a reference object. In 1971, Mean Absolute Differences (MAD) algorithm was proposed by Leese, which was mainly based on the absolute difference between original pixel and the corresponding pixel being used for comparison [12]. Later, Sum of Absolute Differences (SAD) algorithm and Sum of Squared Differences (SSD) algorithm were developed as modified version of MAD [13]. In 1972, Sequential Similiarity Detection Algorithm (SSDA) was illustrated in a paper by Barnea and Sliverman [14]. After a few years, the Sum of Absolute Transformed Differences (SATD) algorithm was proposed as a block matching criterion for video comparison. It worked by taking a frequency transform, usually a Hadamard transform [15].

978-1-5386-5269-5/18/$31.00 ©2018 IEEE

With respect to other features, like color and texture, shape is much more effective in characterizing the edge of an object. Therefore in this paper, based on sum of absolute modules differences, an algorithm is proposed to achieving rotation and scaling parameters as well as independence from the starting point of the edge.

In the design of motion planning part, a method for obstacle avoidance is adopted, which is based on the edge detection and obstacle enveloped. The collision-free assembly is accomplished by increasing the height of the original path.

In regular packaging and assembly production, staff and workers generally do not have detailed knowledge of how to communicate with robots in their language or express the specific task linguistically. To solve this problem, the interactive software system is developed. The design and implementation of GUI is an important part of software system development. GUI refers to the interaction between software system and users. It provides users with various forms of input, including image, data and literal command. After input, the information is transferred to the core module for processing and the processing result will be fed back to the user in an understandable way [16]. In the early 1930s, Vannevar Bush, one of the first people to express the GUI, wrote of a device he called the "Memex" [17]. In about 1974, the first modern GUI, Smalltalk, began to take shape [18]. In 1976, the naissance of Apple's next-generation Lisa computer marked that Steve Jobs became a convert to the GUI religion [19]. In 1985, the first version of Windows was modeled by Microsoft, which was used on the GUI of the Mac OS [20].

In this paper, a graphical user interface (GUI) is designed, which allows real-time object recognition of workspace and collision-free path planning of assembly. The paper is organized as follows. In Section II, a brief description of the general motion planning algorithm underlying the GUI is developed. The algorithm includes FDs method, the image matching algorithm used to accomplish fixed pose assembly and the path planning algorithm. Next, based on the algorithm, the GUI is devised in Section III. This part uses an example to illustrate the details of the function in the GUI. The simulation results in Section IV intuitively show that the GUI can accomplish the assembly task successfully.

II. ASSEMBLY MOTION PLANNING ALGORITHM

A. *Object Recognition and Image Matching*

Starting from the edge of object obtained through image processing algorithm, a discrete-time periodic signal $Z=(z_0,\ldots,z_{N-1})$ is obtained by parameterizing the edge :

$$z_n = x_n + jy_n, \{ n \in [0, \mathrm{N} - 1]; j = \sqrt{-1} \} \quad (1)$$

x_n and y_n are the shape edge coordinates of the ith sampled point. Mapping them to the frequency domain by Fourier transformation, and the Fourier Descriptors (FDs) can be obtained as following:

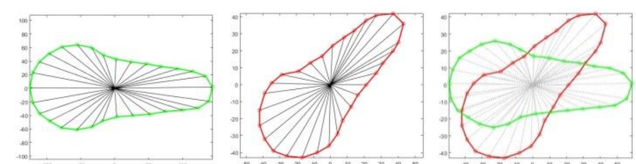
$$F_k = \sum_{n=0}^{\mathrm{N}-1} Z_n e^{(-\mathrm{j}2\pi kn/N)}, \{ k \in [0, \mathrm{N} - 1]; j = \sqrt{-1} \} \quad (2)$$

By modifying the FDs coefficients accordingly (as shown in Table I), the invariance of translation, scaling, rotation, and starting point can be satisfied. And the target objects are recognized with the normalized FDs.

TABLE I. THE NORMALIZED FDS

Geometric invariance	Modified FDs
Translation	$F_0 = 0$
Scale	$F_k = F_k / \|F_1\|, k \in [0, N\text{-}1]$
Rotation and starting point	$F_k = \|F_k\|, k \in [0, N\text{-}1]$

After that, we extract the modules and phase angle of the all edge points in polar space (including reference objects and target objects). The amount of points of reference objects and target objects are resized to be equal. Based on phase angle, all the points are sorted. Taking a rotation transformation on one edge and calculate the sum of absolute modules differences of target object edge points and the reference object edge points used for comparison. The optimal angle of rotation can be obtained when the sum is minimal. Note that in our algorithm, the phase angles of all the edge points remain unchanged. In this algorithm, it is worth pointing out that the computation will not be effected by the choice of start point on edge. In Fig.2, an example is shown to illustrate the details of the algorithm.



(a) (b) (c)

Figure 2. An illustration of the image matching algorithm.

B. *Real–Time Path Planning*

With the algorithm above, the target objects are recognized and the corresponding pose parameters are obtained. In this section, to accomplish the collision-free path planning, we use cylinders with given height to envelop all the items (including objects and obstacles) in workspace and calculate the shortest distance of centroids. As shown in Table II, through comparison of the shortest distance and the radius of cylinders, two types of collision will be detected (obstacle-object in transit and obstacle-end effector). When the collision is detected, instead of generating a new path with interrelated algorithm (such as A*), a more effective strategy is proposed. The algorithm will lift the end effector of parallel robot to increase the height of original path and realize obstacle avoidance.

TABLE II. TYPES OF COLLISION DETECTION

O: obstacle; O_b: object; E: end-effector; D_{min}: shortest distance; R: radius

Collision type	Collision condition
Obstacle-Object in transit	$D_{min}(O,O_b) \leqslant R_O + R_{Ob}$
Obstacle-end effector	$D_{min}(O,E) \leqslant R_O + R_E$

With the adoption of cubic B-spline interpolation, a smooth path of end effector in one work period is shown in Fig.3, where P_0, P_1, P_2 and P_3 represent four points of three processes in the one work period: lifting, translation and descending.

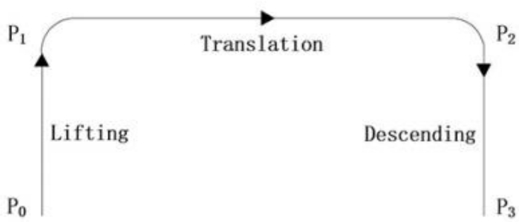


Figure 3. One work period.

III. DEVISE OF THE GUI

Our algorithm enables the parallel robot to accomplish object recognition, obstacle avoidance and assembly path planning. According to the requirements of the practical assembly application, a GUI is devised for users to simulate assembly task, and at the same time it takes full advantage of general features of the proposed algorithm. When started, the general view of GUI is displayed as shown in Fig. 4.

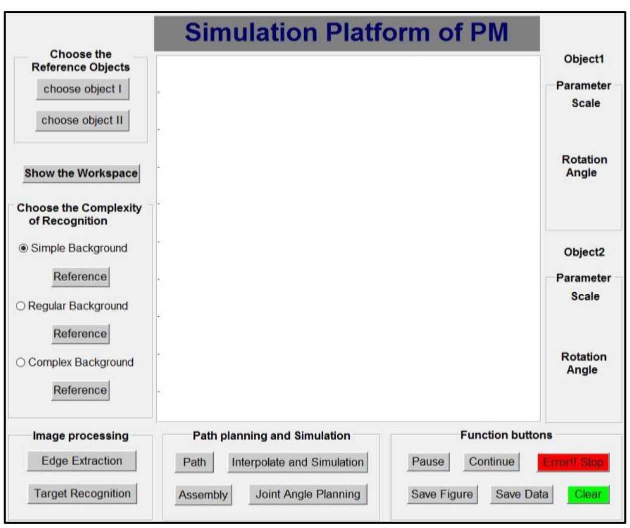


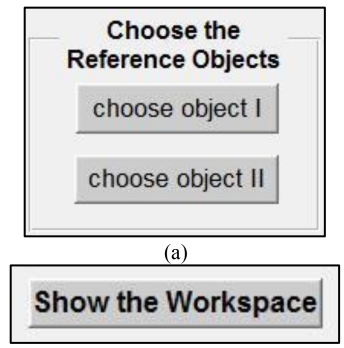
Figure 4. The target workspace shown in the graphic window.

Six button groups, one graphic window and two parameter windows are displayed in this GUI, including reference-object-choice, workspace-choice, image processing, path planning and simulation, object pose parameters windows and function buttons. In this part, we use an example to illustrate the function of the GUI.

A. Choice of Objects and Workspace

As shown in the enlarged view of object-choose button group (Fig.5 (a)), the two reference objects (the objects need to be assembled) can be chosen by pushing buttons respectively. A window of object benchmark will be switched on after push (Fig.6). Users can choose reference object I and II subjectively. And the selected objects will be shown in the graphic window (Fig. 7 (a)-(b)). By pushing the button "Show the Workspace" (Fig.5 (b)), users can enter the workspace benchmark (Fig.8) and choose the target workspace. And then it will be displayed on the graphic window of the GUI (Fig.9).

B. Choice of Background Complexity



(a)

(b)

Figure 5. The object-choose button group.

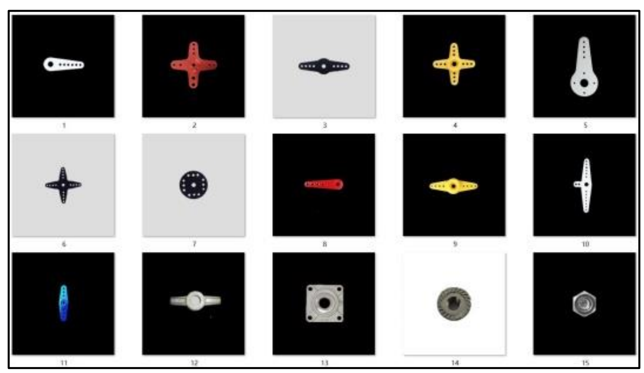
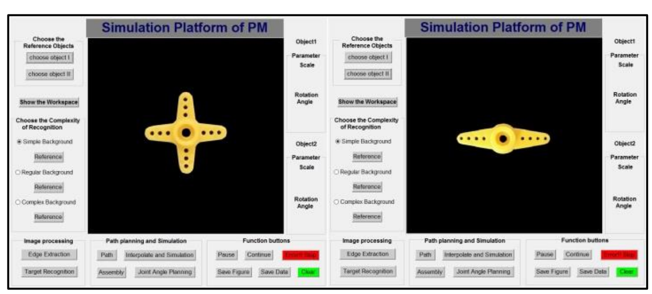


Figure 6. The object-choose button group.



(a) (b)

Figure 7. The reference objects shown in the graphic window.

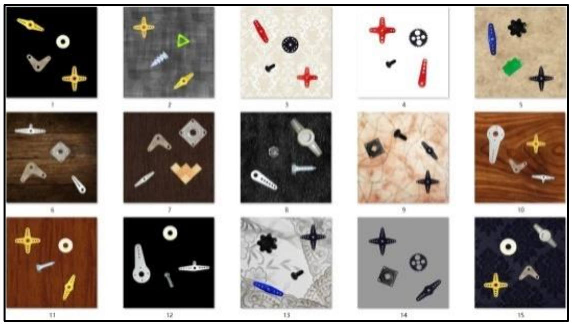


Figure 8. The workspace benchmark.

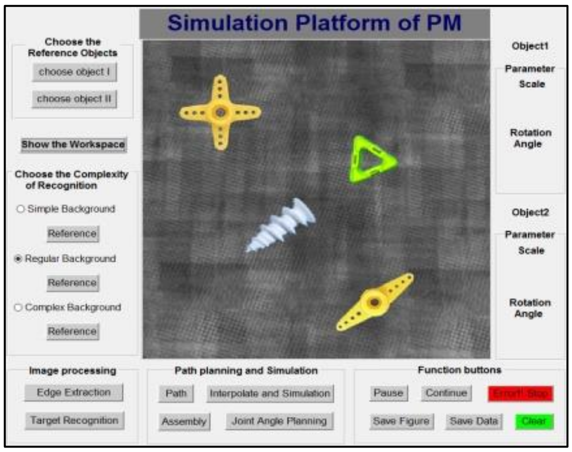
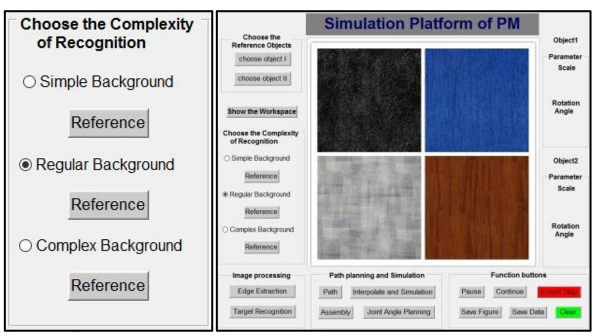


Figure 9. The background-type button group.



(a) (b)

Figure 10. The background-type button group and the regular reference background.

Simple Regular Complex

Figure 11. The reference backgrounds of three types.

According to the regular work conditions in industrial assembly, three types of workspace are provided corresponding to different image processing strategies; they are simple, regular and complex, respectively (Fig.10 (a)). Users can click the radio button "Simple Background", "Regular Background" or "Complex Background" on the basis of the target workspace background, and push the corresponding button "Reference". After that, the reference workspace will be displayed on the graphic window for 3 seconds to instruct users to make a comparison and determine complexity of the selected workspace according to their judgment (Fig.10 (b)). Each type of reference background includes four images for a comprehensive comparison. After 3 seconds, the reference backgrounds will disappear and the selected workspace will be still shown on the graphic window. The reference backgrounds of different types are shown in Fig.11. In this example, the "Regular Background" is chosen.

C. *Image Processing*

Fig. 12 shows the enlarged window of the "Image Processing" button groups. The edges of objects and obstacles can be obtained by pushing the button "Edge Extraction" (Fig. 13). Then the target edges of objects and obstacles can be recognized by pushing the button "Target Recognition", the pose parameters used in assembly (including the pose angle and scale) are also obtained, which are illustrated in the "Parameter" region (Fig. 14). In Fig. 14, where the target object I and II are colored by yellow and red, respectively, while the two obstacles are colored by blue and enveloped by green cylinders. The height of cylinders in workspace is 100mm, which is the maximum height in the workspace.

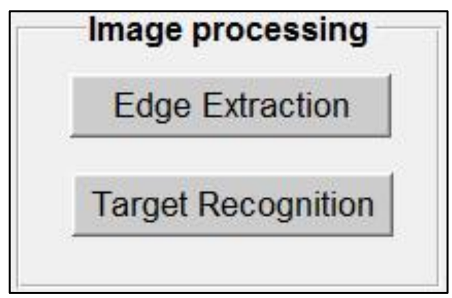


Figure 12. The Image-processing button group.

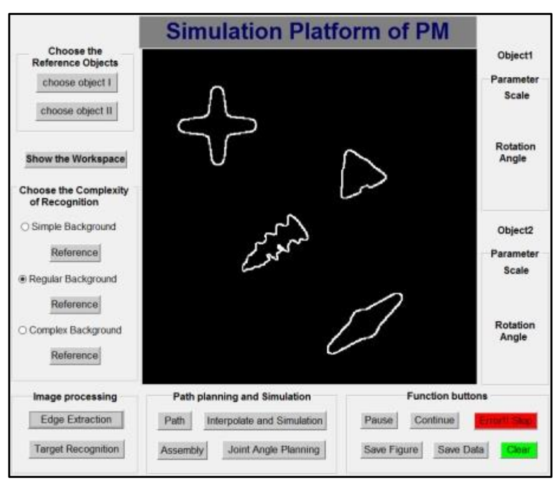


Figure 13. The result of button "Edge Extraction".

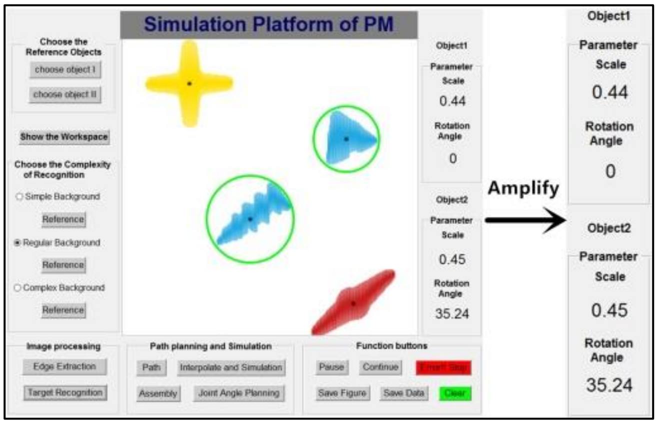


Figure 14. The result of button “Target Detection”.

D. Path-Planning and Simulation

In the stage of assembling motion planning, interrelated buttons are shown in Fig.15. Suppose that the assembling task is to grasp object I (Fig. 7(a)) and move it to object II (Fig.7 (b)), the whole motion of end-effector can be divided into 3 steps (Table III).

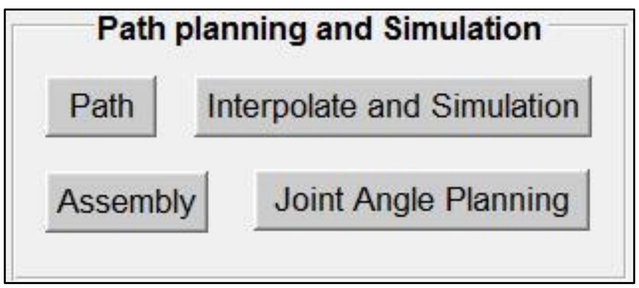
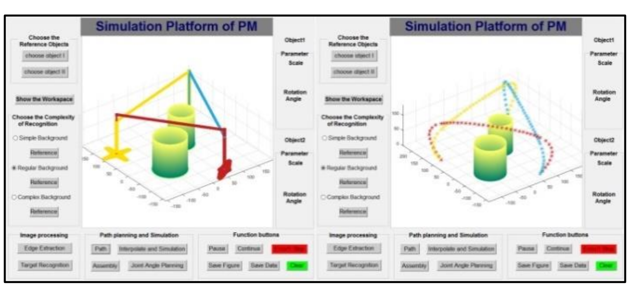


Figure 15. The path-planning button group.

TABLE III. THE MOTION OF END-EFFECTOR.

Step	Motion
I	Move from initial location to object I and grasp it;
II	Deliver object I to the location of object II and accomplish the assembly action;
III	Move back to the initial location;

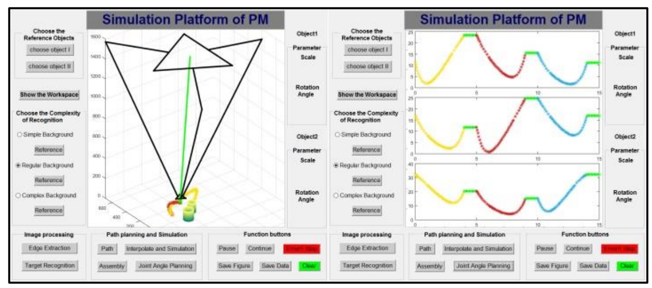


(a) (b)

Figure 16. The results of button “Path” and “Interpolate and Simulation”.

Firstly, users can obtain the point-to-point path by pushing the button "Path"(Fig.16 (a)). The paths of three steps are colored by yellow, red and blue, respectively. In this example, the red path is lifted to realize obstacle avoidance. Then with the button "Interpolate and Simulation", the simulated path of end-effector in whole assembly task is illustrated (Fig.16 (b)). The results indicate that the collision-free assembly task is accomplished successfully.

The button “Assembly” can realize the 3D simulation of assembly process with the application of 4 degree-of-freedom Delta parallel robot (Fig.17 (a)), which including three translational degree-of-freedom (colored by black) and one additional rotational degree-of-freedom (colored by green) for the rotating platform so as to assembly objects in different poses.



(a) (b)

Figure 17. The results of button “Assembly” and “Joint Angles”.

Through the button "Joints Angle Planning", the variation of three joint angles during the three steps (Table III) are illustrated in the graphic window (Fig.17 (b)). Each joint angle curve consists of Step I, Step II and Step III, which are separated by two intermissions of the grasping and releasing action.

E. Function Button Group

Besides the button group illustrated above, another button group of function buttons applied in the GUI is designed to improve the GUI. As shown in Fig.18, five basic function buttons are devised, including “Pause”, “Continue”, “Error !! Stop”, “Save Figure”, “Save Data”. “Pause” and “Continue” are applied in the simulation of assembly for observation and detection. “Error !! Stop” is used in the GUI to make provision against emergencies. “Save Figure” and “Save Data” are designed to save the images and data of results.

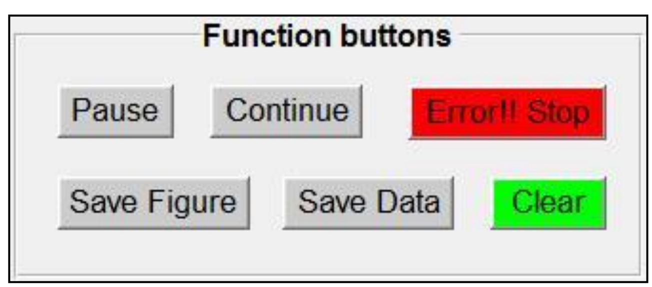


Figure 18. The function button group.

IV. SIMULATION

In this section, we have conducted several applications of our proposed GUI in MATLAB. To obtain a clearer observation, only the graphic window is shown. The results of the applications are shown in Fig. 19-22.

A. Application I   *The choice of complexity: Simple*

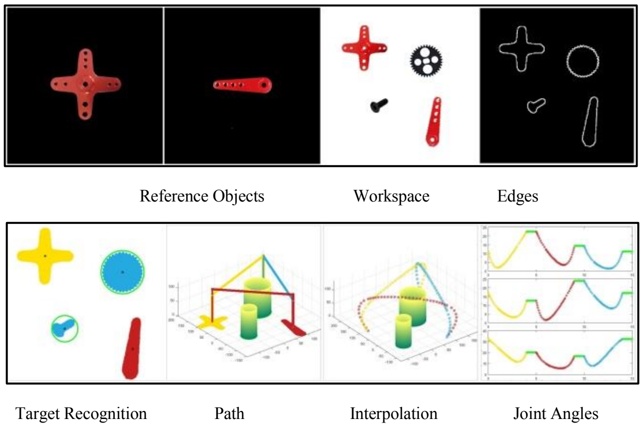


Figure 19. The results of application I.

B. Application II   *The choice of complexity: Regular*

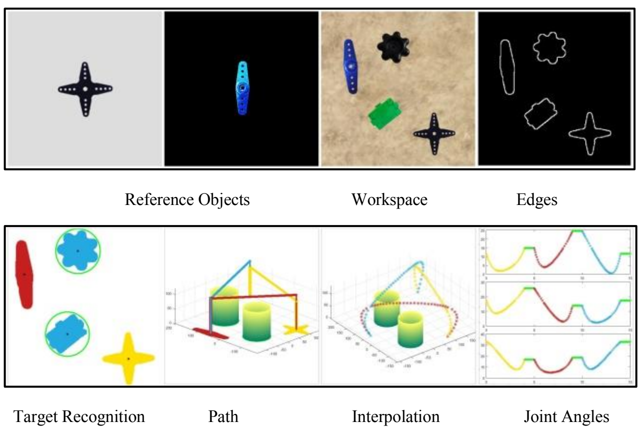


Figure 20. The results of application II.

C. Application III   *The choice of complexity: Complex*

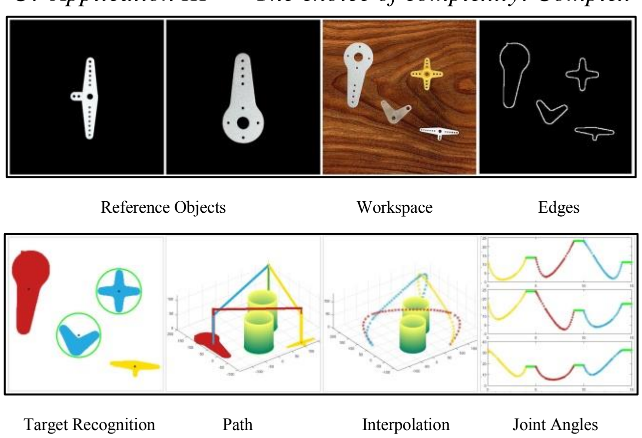


Figure 21. The results of application II.

D. Application IV   *The choice of complexity: Complex*

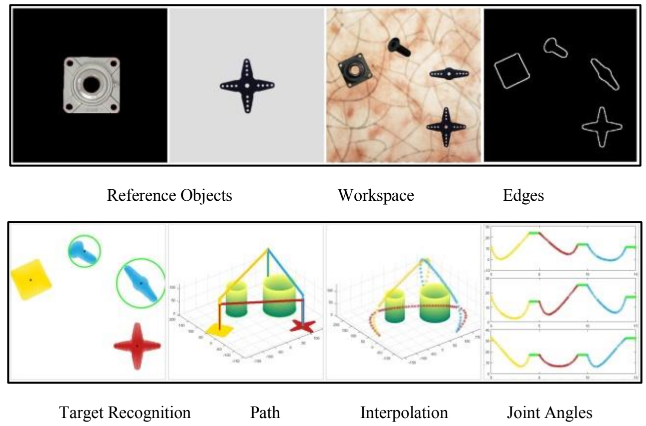


Figure 22. The results of application IV.

The results of applications show the effective and accuracy of the algorithm and the practical of the proposed GUI.

V. CONCLUSION AND FUTURE WORK

The collision-free motion planning for the assembly task of Delta parallel manipulator is a spatial problem. For this reason we have devised a GUI that enables users to realize reference objects and workspace selection, target detection, path planning and have a simulation in real-time. Users need no prerequisites in mathematics, robotics or machine vision. The conserve of figure and data is also possible and easily commanded through the GUI. Applications of the GUI show the practical and feasible of it. And through further optimization, the GUI will be applied in the industrial production in the near future.

ACKNOWLEDGEMENT

This work is supported by National Science Foundation of China (Project No.51775155 and 61502135).

REFERENCES

[1] Vischer P, Clavel R. Kinematic calibration of the parallel Delta robot [J]. Robotica, 1998, 16(2): 207-218.

[2] Wittenburg J. Stewart Platform [M]. Kinematics. Springer Berlin Heidelberg, 2016.

[3] Gallardo-Alvarado J. Delta Robot [M]. Kinematic Analysis of Parallel Manipulators by Algebraic Screw Theory. Springer International Publishing, 2016.

[4] Gallardo-Alvarado J, Balmaceda-Santamaría A L, Castillo-Castaneda E. An application of screw theory to the kinematic analysis of a Delta-type robot [J]. Journal of Mechanical Science & Technology, 2014, 28(9):3785-3792.

[5] Gutiérrez-Preciado A, González-Palacios M A, Aguilera-Cortés L A. Workspace Analysis of a Delta-Like Robot Using an Alternative Approach [M]. Multibody Mechatronic Systems. Springer International Publishing, 2015, 25: 453-463.

[6] Ponce, Jean. Computer vision :[M]. Publishing House of Electronics Industry, 2012.

[7] Amlekar M M, Gaikwad A T, Manza R R, et al. Leaf shape extraction for plant classification [C]. International Conference on Pervasive Computing. IEEE, 2015:1-4.

[8] Vanishree, Reddy K V R. Implementation of pipelined sobel edge detection algorithm on FPGA for High speed applications [C]. International Conference on Emerging Trends in Communication, Control, Signal Processing & Computing Applications. IEEE, 2014:1-5.

[9] Sangston K J, Gini F, Greco M S. Coherent Radar Target Detection in Heavy-Tailed Compound-Gaussian Clutter[J]. IEEE Transactions on Aerospace & Electronic Systems, 2012, 48(1):64-77.

[10] Aprovitola A, Gallo L. Edge and Junction Detection Improvement Using the Canny Algorithm with a Fourth Order Accurate Derivative Filter [C]. Tenth International Conference on Signal-Image Technology and Internet-Based Systems. IEEE, 2015: 104-111.

[11] El-Ghazal A, Basir O, Belkasim S. Farthest point distance: A new shape signature for Fourier descriptors [J] Signal Processing Image Communication, 2009, 24(7):572-586.

[12] Zhu H G, Feng Q. A Video Bit Rate Control Algorithm Based on Weighted Estimate of Image Brightness Difference [J]. Applied Mechanics & Materials, 2013, (347-350): 844-848.

[13] Dang K H, Le D, Dzung N T. Efficient determination of disparity map from stereo images with modified sum of absolute differences (SAD) algorithm [C]. International Conference on Advanced Technologies for Communications. IEEE, 2014: 657-660.

[14] Chen H, Ma G, Wang Z, et al. Probabilistic Detection of Missing Tags for Anonymous Multi-Category RFID Systems [J]. IEEE Transactions on Vehicular Technology, 2017, PP(99): 1-1.

[15] Zhao C, Lee W S, He D. Immature green citrus detection based on colour feature and sum of absolute transformed difference (SATD) using colour images in the citrus grove [J]. Computers & Electronics in Agriculture, 2016, 124(C): 243-253.

[16] Cimolini P. GUI Design [M]. Oracle Application Express by Design. 2017.

[17] Lidierth M. GUI History automatically generates MATLAB code based on a user's actions in a GUI [J]. Organometallics, 2010, 29(5): 1210-1221.

[18] Cimolini P. GUI Design[M]. Oracle Application Express by Design. 2017. https://arstechnica.com/features/2005/05/gui/

[19] Louridas P. Declarative GUI Programming in Microsoft Windows [J]. IEEE Software, 2007, 24(4): 16-19.