Motion Planning and Control of a Picture-Based Drawing Robot System

Chun-Fei Hsu
Dept. of Electrical Engineering, Tamkang University, New Taipei City 25137, Taiwan
e-mail: fei@ee.tku.edu.tw

Wei-Heng Kao
Dept. of Electrical Engineering, Tamkang University, New Taipei City 25137, Taiwan
e-mail: jackgao0323@gmail.com

Wei-Yu Chen
Dept. of Electrical Engineering, Tamkang University, New Taipei City 25137, Taiwan
e-mail: dodie_win@hotmail.com

Kai-Yi Wong
Dept. of Electrical Engineering, Tamkang University, New Taipei City 25137, Taiwan
e-mail: 400500111@s00.tku.edu.tw

***Abstract*—The traditional drawing robot applications mostly need lots of manpower to select plotting points and determine the drawing process, which would took a long time and hard to design. To solve this problem, this paper proposes a drawing robot system, which can draw a picture what user wanted. There are two main parts in the drawing robot system. One is the delta robot manipulator that includes inverse kinematics and motor position control, and the other is image processing which detect edges and do motion planning for an image. Finally, the experiment results illustrate that the drawing robot system can save enormous labor resources and time by executing automatically. The detailed contributions of this paper are listed as follows: (1) the drawing robot system executes more efficiently compared to previous works. (2) The drawing robot system can save enormous labor resources and time by executing automatically. (3) The drawing robot system provides the results with high integrity.**

***Keywords*—delta robotic arm; edge feature; drawing system.**

I. Introduction

In automatic industrial automation, manufacture industry field, there are wide field application on robot manipulator which can do precise manufacture, high repeated movement property. This can improve quality and efficiency on manufacture. But it needs programming to drive robot manipulator moving, and needs professional engineer in the field to control robot manipulator moving [1, 2]. As technology advances, the competition of society is more and more fierce. In order to reach the maximum efficiency, a lot of companies start using robots to replace manpower and most common is robot arms which can be divided into parallel robot arm and series robot arm by the mechanical structure.

The traditional drawing robot applications mostly need lots of manpower to select plotting points and determine the drawing process, which would took a long time and hard to design. In recent years, many drawing robot systems have been proposed [3-6]. In [3], Hung proposed "The Basic Technology Research of Robot Arm Drawing System" which user can draw on a computer window and the series robot arm can draw the same thing on the platform. However, it needs human to determine feature points of the curve line. In [4], Hámori et al. proposed "LEGO NXT-based Robotic Arm" by using inverse kinematics to derive the moving path, but the structure of the robotic arm is not stable. In [5], Shen proposed "Drawing System for Six-axis Robot Manipulator based on B-Spline Curves". Although the system can achieve an excellent result, it needs lots of time and manpower. In [6], Li proposed "Picture-based Drafting System for Robot Manipulators". The path planning combines image processing instead of the time consuming which draw by human, but the six-axis robot arm is very expensive.

Generally, it needs developer to choose drawing points, edit robot manipulator drawing path and edit robot manipulator moving path all by manual in traditional drawing method. According to above, there are two main time consuming part : (1) Choose robot manipulator moving points by manual, and (2) Programming on control robot manipulator moving. Thus, drawing picture quality and develop speed depend on the developer ability.

This paper proposes a drawing robot system, which can draw a picture out what user wanted, as shown in Fig. 1. We use image processing algorithm to implement drawing robot system which can generate drawing path and robot manipulator moving path. The parallel robot arm is designed as the robot arm and TM4C123GXL LaunchPad is chosen as the main core. The software uses Code Composer Studio to derive the formula of the parallel robot arm by using inverse kinematics. Thus, this paper uses image processing to identify the information, detect edge and find out the feature points of the image. After getting the information, we define the coordinate and do the path planning. Control the parallel robot arm to complete the drawing mission.

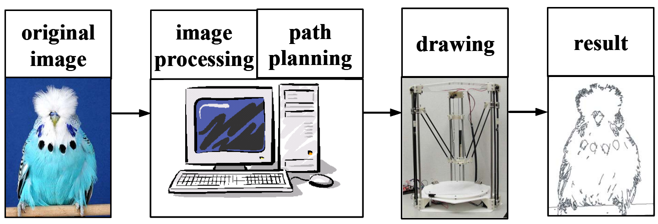


Fig. 1. The proposed drawing robot system.

The authors appreciate the partial financial support from the Ministry of Science and Technology, Taiwan, under Grant MOST 103-2221-E-032-063-MY2.

978-1-5090-4917-2/17/$31.00 ©2017 IEEE

II. DRAWING ROBOT SYSTEM

There are two important parts in the drawing robot system we proposed. One is the delta robot manipulator that includes inverse kinematics and motor position control, and the other is image processing which should detect edges and motion planning of an input image.

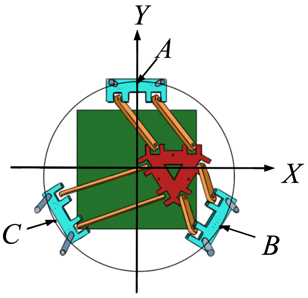


Fig. 2. The position schematic of each pillar on Cartesian coordinate.

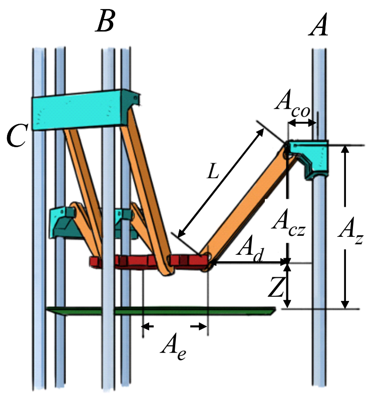


Fig. 3. The definition schematic of each structure.

A. Delta Robot Manipulator

Inverse kinematics is very important in robotics because robot manipulator through the joint angle to control. The concept of inverse kinematics is deriving the angle change of the robot arm's each joint through a desired position of the end-effector. In this paper, define A pillar on the Y axis and define B and C pillar on the position that rotate 120 degrees separately, as shown in Fig. 2. Then we need to define every joint of the parallel robot manipulator. Each base of motor connects with two plain shafts. Each pair of plain shaft has two slides on it and moving through the belt. Slides and platform of the robot arm are connecting with the carbon stick and universal joint as shown in Fig. 3, where A_{co} is the horizontal distance between A pillar and the carbon stick, A_e is the horizontal distance between the center in the platform of the robot arm and the point connect with carbon stick and the platform, A_z is the vertical distance between the slide of A pillar and the platform of the structure, A_{cz} is the vertical distance between the slide and the platform of robot manipulator, and Z is the vertical distance between two platforms. Therefore, we can get an equality equation as following

$$Z = A_z - A_{cz} = B_z - B_{cz} = C_z - C_{cz} \tag{1}$$

Thus, we can derive the final equations of the parallel robot arm as following

$$A_{cz} = \sqrt{L^2 - X^2 - (Y - DR)^2} \tag{2}$$

$$B_{cz} = \sqrt{L^2 - (X - \frac{\sqrt{3}}{2}DR)^2 - (Y + \frac{1}{2}DR)^2} \tag{3}$$

$$C_{cz} = \sqrt{L^2 - (X + \frac{\sqrt{3}}{2}DR)^2 - (Y + \frac{1}{2}DR)^2} \tag{4}$$

where DR is the horizontal vector of the carbon stick when the center of the robot manipulator overlap with the center of Cartesian Coordinate System. From above equations we can obtain that when give a coordinate command (X, Y) thus the command of each motor can be got by using (2)~(4).

B. Image Processing

OpenCV (Open Source Computer Vision Library) is initiated and participating the development by Intel, that using "Berkeley Software Distribution license" to authorize the publication and it can use in business and research areas for free. It is a cross-platform computer vision library that includes hundreds of algorithms. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision. In order to draw the picture, we use Canny Edge Detection to find the maximum gradient of pixels to get the marginal point and it can calculate the edge of the picture. Canny Edge Detection which was developed by John F. Canny is a popular edge detection algorithm. It owns the advantage of detecting all edges of pictures in effect, excellently locate ability, the error of detection is very small, and the point on the edge have good reaction. The edge detection results for image no. 1 and 2 are shown in Figs. 4 and 5, respectively. It shows that we can get is strong edges in the image by using a single function in OpenCV Library.

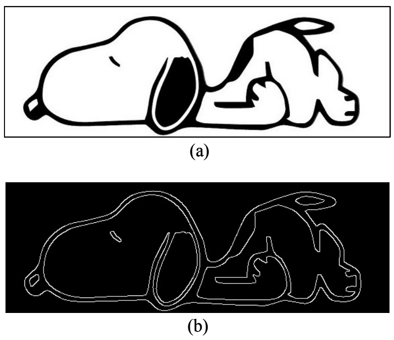
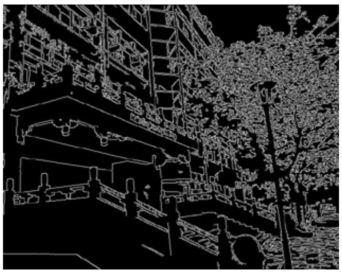


Fig. 4. Edge detection result for image no. 1. (a) original image. (2) detection result.

(a)



(b)

Fig. 5. Edge detection result for image no. 2. (a) original image. (2) detection result.

III. Path Planning

A. Simple Path Planning

At the beginning, the method we use to do the path planning is to draw every pixel. We scan every pixel of the picture from the top left corner in horizontal direction. If we scan white point, we move the pen to the position to draw and record then we scan the nearby points. If no white point nearby, we lift the pen and scan other white point. As shown in Fig. 6, we draw the picture according to this simple path planning method. Though this method is easy and achieve easily, it has a fatal error which is if the white point we scan is not the endpoint of a line, it will cause a line divide into many pieces. It will suffer from a heavy computation loading to finish the drawing task.

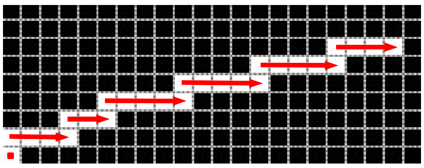


Fig. 6. Simple path planning method.

B. Improved Path Planning

As shown in Fig. 6, we can find that the line is not continuous. To attack this problem, this paper proposes an improved path planning method. After the image through the image processing, we gave each pixel a definition as shown in Fig. 7. Our paper divides every pixel into three types. First type is start point/end point, second type is branch point and the third type is normal point. Start point/end point is the beginning/end of a line, branch point needs to determine how many times the path will pass by to complete the whole picture, and other points are normal point.

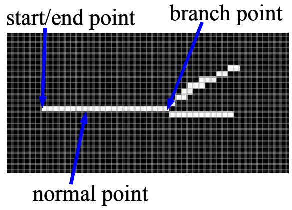


Fig. 7. The type determination of the pixel.

To determine which type the pixel is, we use the pixel as the center point to observe the nearby points. As shown in Fig. 8, we determine the start point/end point as a beginning/end of a line, so it only has one way can pass. After we discuss the normal point which is a process of a line, the different between the normal point and start point/end point is the normal point has two ways can pass. As shown in Fig. 9, we divide normal point into five situations as 2, 3, 4, 5 and 6 white points nearby. These are all situation of the normal points. Last, the branch point is more special than normal point and start point/end point. As shown in Fig. 10, there are all situation of branch point. The rule we choose path of normal point and branch point is we randomly choose a nearby point that is on the cross of the target center point and find the next point that separated by 90, 120 or 180 degrees. If it only has one path we define the point as normal point, if it has more than one way we define it as branch point.

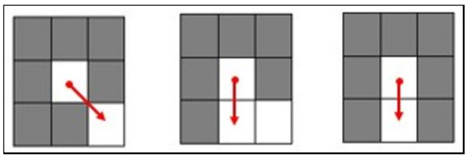


Fig. 8. Schematic of start/end point.

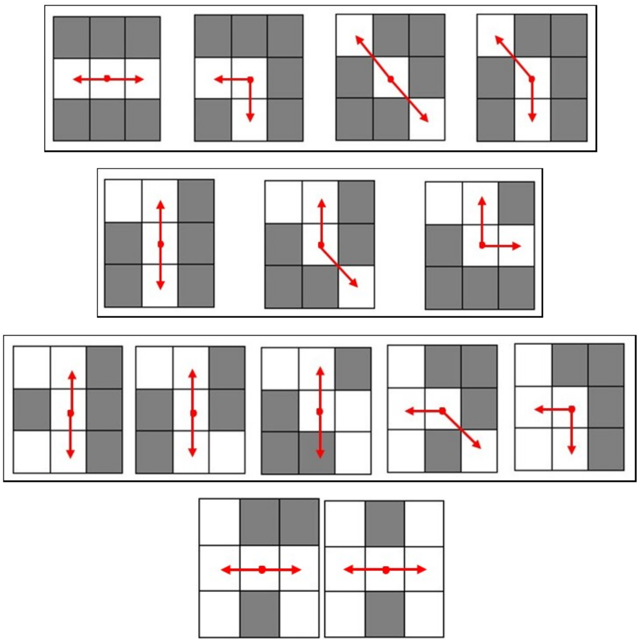


Fig. 9. Schematic of normal point.

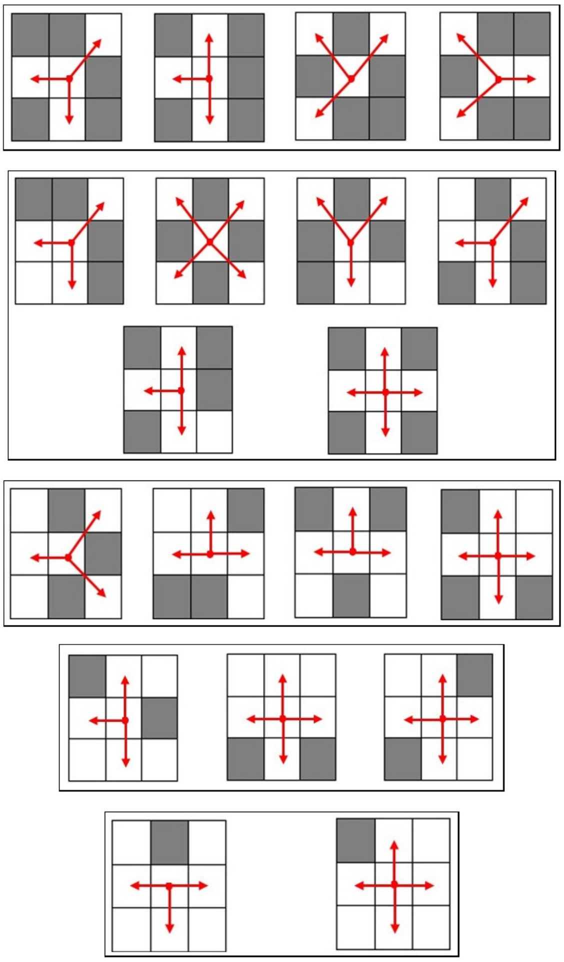


Fig. 10. Schematic of branch point.

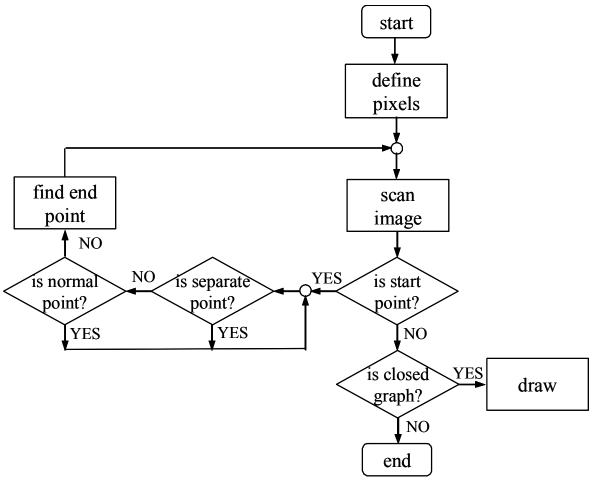


Fig. 11. Flowchart of the improved path planning.

The flowchart of the proposed improved path planning is shown in Fig. 11. In drawing processing, the last situation that our paper considers is the closed graph. We can find the graph that it don't have a start point/end point, so after finish the step above. After we draw the remaining of closed graph, we finish the whole picture. By the skill of improved path planning, not only can solve the disadvantage of breakpoint but also can speed up the drawing speed to have a better result.

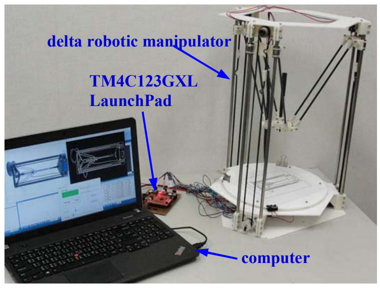


Fig. 12. The picture of the proposed drawing robot system.

IV. Experimental Results

The traditional drawing robot applications mostly need lots of manpower to select plotting points and determine the drawing process. The developed drawing robot system is shown as Fig. 12. The feature points can be obtained by using canny algorithm via OpenCV approach and the routing path is calculated by using a motion planning algorithm. There are two main parts in the drawing robot system, one is the delta robot manipulator and the other is image processing. We use TM4C123GXL LaunchPad that produced by Texas Instrument to control the delta robot manipulator with inverse kinematics and we use computer to implement the image processing and path planning. To show the effectiveness of the proposed drawing robot system, two scenarios are tested here. One is the black-and-white image and the other is the colorful image. The experimental results of scenarios 1 and 2 are shown in Figs. 14 and 15, respectively. It shows that the proposed improved path planning method not only solves the disadvantage of breakpoint but also speed up the drawing speed.

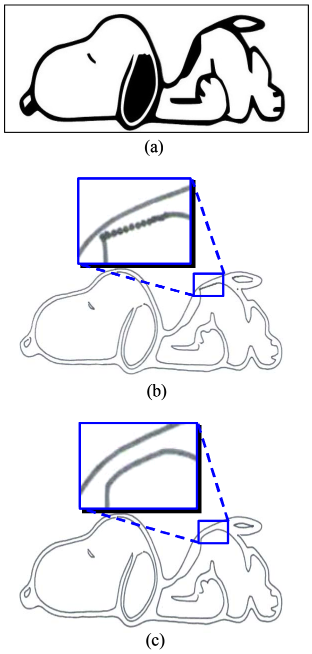
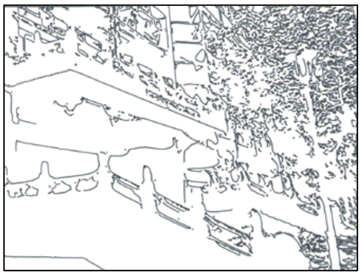
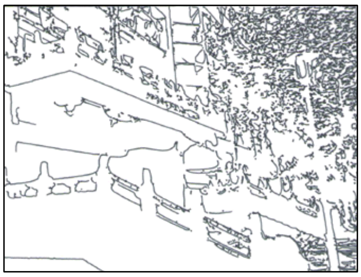


Fig. 13. The experimental results of scenario 1. (a) original image. (b) using simple path planning. (c) using improved path planning.

(a)



(b)



(c)

Fig. 14. The experimental results of scenario 2. (a) original image. (b) using simple path planning. (c) using improved path planning.

V. CONCLUSION

There are two main parts in the proposed drawing robot system. One is the delta robot manipulator and the other is image processing. In the first part, TM4C123GXL LaunchPad is used as the main control board which includes inverse kinematics and motor position control. In the second part, the feature points can be obtained by using canny algorithm via OpenCV approach and the routing path is calculated by using a motion planning algorithm. The process of Canny edge detection algorithm in OpenCV can find the boundaries of objects within images. In additional, an improved path planning method is proposed to speed up the drawing speed and to achieve favorable drawing results. From the real-time experimental results, it can show that the proposed drawing robot system have advantages of fast, simple and save manpower and lots of time. The experiment results illustrate that the drawing robot system can save enormous labor resources and time by executing automatically.

ACKNOWLEDGMENT

The authors appreciate the partial financial support from the Ministry of Science and Technology, Taiwan, under Grant MOST 103-2221-E-032-063-MY2.

REFERENCES

[1] M. Fujita and H. Kitano, “Development of an autonomous quadruped robot for robot entertainment,” Autonomous Robots, vol. 5, pp. 7-18, 1998.

[2] G. J. Pierre and Z. Said “The artist robot a robot drawing like a human artist,” IEEE Conference on Industrial Technology, pp. 486-491, 2012.

[3] C. Y. Hung, The basic technology research of robot arm drawing system, Master thesis, National Taiwan University of Science and Technology, Taiwan, 2007.

[4] H. Ákos, L. János, and R. Barna, “3DOF drawing robot using LEGO-NXT,” 15th International Conference on Intelligent Engineering Systems, pp. 293-295, 2011.

[5] Y. J. Shen, Drawing system for six-axis robot manipulator based on B-spline curves, Master thesis, Tamkang University, Taiwan, 2013.

[6] G. T. Li, Picture-based drafting system for robot manipulators, Master thesis, Tamkang University, Taiwan, 2015.