Project-Using robot arm to draw human portraits

Yue Yiyao, Wang Yexiang, Shi Yongqi

I. INTRODUCTION

Robot painting refers to a form of art that uses robot systems to create paintings. Robot painting can be created entirely by robots themselves, or it can be completed under the guidance and participation of humans.

Robot manipulators have been widely used in modern society by human beings for a number of years, for industrial production in particular. But with the rapidly developing science and technology, robot arms are gradually used in cases of recreation and art. Robot drawing is obviously one of the best examples of this case.

The traditional drawing robots are not so intelligent. They require a big amount of manpower to determine the spots and specify the detailed process when they draw. Apart from this, they take a lot of time to complete a drawing.

On the other hand, there have being more and more research on this field and drawing robot manipulators with better automation and intelligence developed and used, which we are going to study and use for reference.

In robot painting, robots are usually equipped with various sensors and cameras to perceive changes in the surrounding environment and painting process. At the same time, robots will also be equipped with painting tools equipped with ink or crayons of different colors, as well as precise painting equipment and motion control systems, enabling accurate drawing of complex patterns and lines.

The advantage of robot painting is that it can achieve efficient production and high-quality output of creativity. Because robots have the characteristics of high precision, high speed, and sustainable and stable operation, they can quickly complete large-scale painting creations in the field of painting art, and can complete complex painting techniques that are difficult for humans to achieve.

II. OBJECT OF THE PROJECT

A robot arm with a camera is positioned in front of a table, as shown in Figure 1. Its vision system detects the presence of a user facing the robot. Upon detecting the user's approach, the robot turns its head towards the user and asks whether the user would like to have a portrait drawn by the robot, using its speech synthesis system. If the user confirms their interest, the robot captures an image of the user's face and with the pen in its grasp, the robot returns its gaze to the table and commences sketching the user's portrait naturally, as if drawn by a human. Starting with rough contours and iteratively adding details, the robot arm fills in dark areas

using different styles of shading. As the ink in the pen runs low, the robot refills the pen with ink from an inkpot. Once the portrait is complete, the robot signs the drawing. The sequence of events is illustrated in Figure 2.

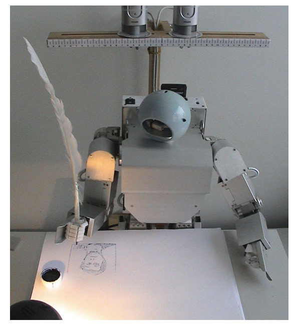


Figure 1.

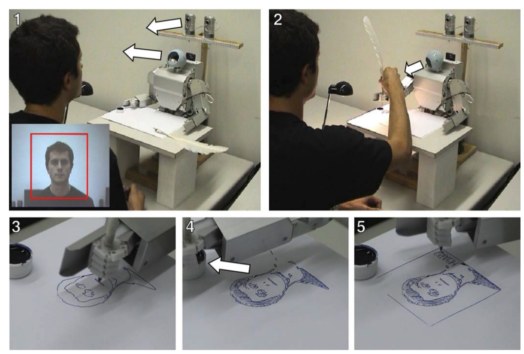


Figure 2. Experimental scenario.

1) The vision system detects the presence of a user (see inset), takes a snapshot of his face, while the robot turns its head toward the user. 2) The robot moves its arm toward the user and asks for its quill pen, using speech synthesis. Speech recognition is used to give simple instructions to open and close the robot's hand, so that it can grasp the pen hold by the user. The robot grabs the pen, looking at its hand. 3) The robot starts sketching the portrait of the user, starting with the rough contours, and adding details iteratively. 4) It then fills the areas bounded by the contours. When the quill pen is empty, i.e. when the length of the trajectory has reached a given threshold, the robot uses an ink-pot to refill the pen. 5) Finally, the robot draws a frame around the portrait, signs its drawing, turns its head to the user, and signals verbally that the portrait is finished.

III. REVIEW

Some products in market

AxiDraw: AxiDraw is a desktop robotic drawing machine that can draw high-precision lines, curves, and text, supporting drawing on various flat surfaces such as paper, fabric, leather, etc. https://www.evilmadscientist.com/2016/axidraw/

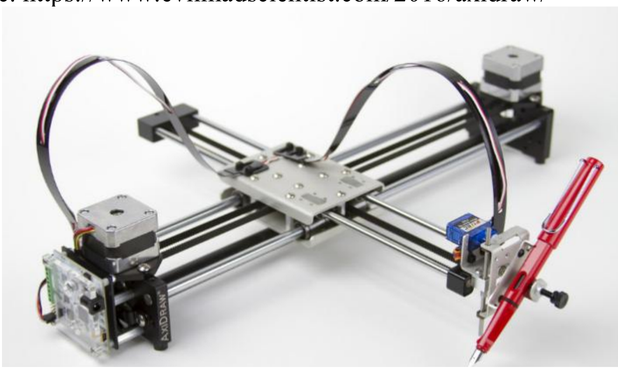


Figure 3 AxiDraw

uArm Swift Pro: The uArm Swift Pro is a desktop robotic arm with an open structure that can be programmed and controlled to write and draw. By connecting to a computer and programming software, users can use the uArm Swift Pro for writing, drawing, line drawing, and more.

Figure 4 uArm Swift Pro

Line-us: It is a small drawing robot arm that connects to your phone, tablet, or computer via Wi-Fi. You can draw anything on your device, and Line-us will mimic your drawing on paper. You can also save and share your drawings via the Line-us app.

Figure 5 Line-us

ZKM： ZKM laboratory created the first real-time robot portraitist system, in which an industrial robot drew the face portrait for the human sitting in front of the camera [2].

Method

In the robot project that called Pica[3], we analysis its method.

The image algorithm will first use the face detection function to locate the human face, then use the center-off techniques [3][4] to attain the representative curves and line segments on the face, and next use the median algorithm to remove unwanted noises. After that, the facial feature processing techniques [5] are implemented to locate eyebrows, eyes, nose, and mouth. The hair regions are located by using the segmentation techniques. The final line segments representing the face portrait are recorded and the coordinates of the key points on these line segments are sent to the arm motion control algorithms to prepare the portrait painting.

The Center-off algorithm [3][4] is used to transform the full face image into a sketch-like face description consisting of a number of characteristic line segments that show most representative figures of the face. Common used edge detection algorithms such as Sobel edge detector have drawbacks on determination of thresholding values. As shown in Fig. 6, the original image before undergoing the Center-off algorithm and the one after being transformed into a clean face portrait consisting of important and characteristic line segments.

Figure 6

At the end of the image processing, a large number of point coordinates representing the human face portrait are recorded as in the 2D(x,y) drawing paper coordinate system. These coordinates will be transformed to the global 3D world coordinate (x,y,z) system for the robot. Based on the 3D coordinates, the joint angle trajectory of Pica's drawing arm is computed. Three main methods are often used to obtain the inverse kinematics solutions, the closed form, the geometric and the numeric solutions. The closed form (algebraic form) and the geometric inverse kinematics are significantly more efficient than the numeric solutions. But the geometric form inverse kinematics solutions become complicated for Pica's arm because it has the angle (o15) at the first joint. Therefore, the closed form inverse kinematics is used in this case. Fig.7 shows the configuration of Pica's 7 DOF arm and its drawing pen, and also the representation of the motor distribution, link lengths, and the local coordinate systems.

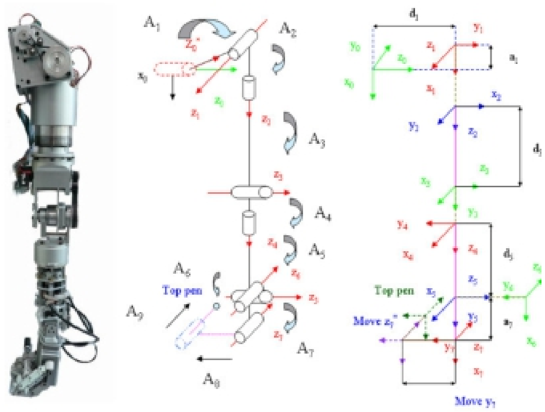


Figure 7

A number of people have been used on the real-time robot portrait drawing experiments. As long as the illumination conditions and the face restrictions are met, the real-time portrait drawing success rate is quite high. Fig. 8 shows the CCD captured image, the image process algorithm generated image and the robot drawn portrait for 3 people. It took 4-6 minutes to finish the portrait drawings.

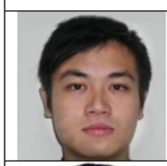
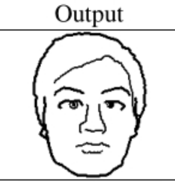
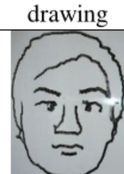
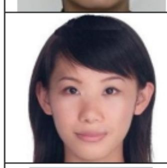
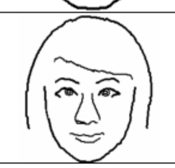
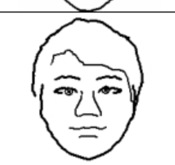
Input Image	Program Output	Robot drawing
		
		
		

Figure 8

There are still other methods:

In the beginning, we need to detect the target image and do some processing, in order to make the picture more suitable and easier for the robot manipulator to draw. So we need to detect the edges, or contours, of the image taken by the camera. In this way, the robot arm can recognize the lines and curves it has to draw on the paper. To realize this function, we can use Canny Edge Detection, which is an effective algorithm to extract the edges from a picture. This is an algorithm proposed in 1986 by John F. Canny, and it is widely used in research even if it is an old method with a long history.

Canny Edge Detection can improve the accuracy of edge detection, which means that the possibility of miss detection and wrong detection is very small. In addition, it can reduce the influence of the noise, and accurately locate the edges.

The procedures of the Canny Edge Detection algorithm can be divided into 4 parts, Gaussian filtering, gradient computation, non-maximum suppression, and double thresholds detection. The Gaussian filtering in the first step is for noise reduction, in order to avoid the wrong detection. The gradient is used to find out the possible edges, and the non-maximum suppression can detect the local maximum value, which generates thin edges. And the 2 thresholds, which should be determined by the user, are for estimating the possibility of whether what it detects is a real edge.

The Canny Edge Detection algorithm can be called from OpenCV(Open Source Computer Vision) library.

After generating the edges for the robot manipulator to draw, we need to decide how to draw the pictures for the robot, which corresponds to the problem of path planning. A simple solution to this problem is that we can scan the whole image pixel by pixel. If we find the edge, which is represented in white pixels, the manipulator controls the pen and moves it down to draw it. Then the robot scans the nearby pixels and draws other white pixels if there are any, otherwise, it lifts the pen can continue to find other white pixels in the picture. This method is simple and easy to implement, but it may divide a line into many small pieces, which may not lead to a good performance. Meanwhile, the robot arm needs to drop and lift the pen many times, which will take a lot of time.

For the problem above, we can choose an improved path-planning method. In this case, we divide the white pixels into 3 groups, the start points or end points, the branch points, and the normal point, instead of just simply determining whether the current pixel is a part of the edge. The start or end points have only one way of white pixels out from themselves, while the normal points and the branch points have 2 and 3 ways out, respectively. We need to find out all the cases for the 3 groups of points, we should start with the start or end points when drawing, in order to produce a smooth line.

After the curves and lines are detected, and the path planning is completed, we can describe the paths for the robot manipulator by Freeman chain code. Freeman chain code is often used in image processing, and computer graphics to represent curves and edges. In a digital image. Each pixel has 8 neighboring pixels around it. This also means 8 directions which can be described by 8 numbers. For each pixel, we can choose one of the 8 numbers from 0 to 7 to represent the direction of the next pixel. When every pixel is assigned a number, the robot manipulator can use these numbers to draw the curves.

IV. Preliminary design scheme

1. Control system of the robotic arm: The control system of the robotic arm should be able to precisely control the movement of the arm, so that it can accurately draw the portrait. In this project, the Gen3_lite will be used directly and use its own control program in lab.

2. Facial recognition technology: In order to accurately draw the portrait on paper, facial recognition technology needs to be used to identify facial features such as the contour of

the face, eyes, nose, and mouth, and then use these features to draw the portrait.

3. Drawing techniques: The robotic arm needs to be able to use drawing tools (such as a pen or brush) to draw, while also considering the use of different colors and line thicknesses to add detail and depth to the portrait.

4. Algorithm and programming design: To achieve the drawing of the portrait, algorithms and programs need to be written to control the movement of the robotic arm, as well as to combine facial recognition technology with drawing techniques, so that the robotic arm can automatically draw the portrait.

REFERENCES

[1] S. Calinon, J. Epiney and A. Billard, "A humanoid robot d rawing human portraits," 5th IEEERAS International Confe rence on Humanoid Robots, 2005., Tsukuba, Japan, 2005, p p. 161-166, doi: 10.1109/ICHR.2005.1573562.

[2] Robot-lab Art, http://www.robotlab.de/index.htm

[3] Chyi-Yeu Lin, and Chien-Wei Li, " Human Portrait Generation System for Robot Arm Drawing, " The 13th Conference on Artificial Intelligence and Applications. Taipei, Taiwan, 2008.

[4] Xiaoyang Mao, Yoshiyasu Nagasaka and Atsumi Imamiya, "Automatic Generation of Pencil Drawing From 2D Images Using Line Integral Convolution," CAD/Graphics , 2001.

[5] Shigefumi Yamamoto, Xiaoyang Mao and Atsumi Imamiya, "Enhanced LIC Pencil Filter," IEEE Computer Graphics, Imaging and Visualization Proceedings, 2004.

[6] M. Pichkalev, R. Lavrenov, R. Safin and K. - H. Hsia, "Face Drawing by KUKA 6 Axis Robot Manipulat or," 2019 12th International Conference on Developments i n eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 7 09-714, doi: 10.1109/DeSE.2019.00132.