

Robot Artist - Automated Picture Portrait

Abdulrahman Alkhodairy

Robotics, Intelligent Sensing & Control (RISC)
Laboratory,
School of Engineering, University of Bridgeport,
221 University Avenue, Bridgeport, CT 06604,
USA.
aalkhoda@my.bridgeport.edu,

Sarosh Patel

Robotics, Intelligent Sensing & Control (RISC)
Laboratory,
School of Engineering, University of Bridgeport,
221 University Avenue, Bridgeport, CT 06604,
USA.
saroshp@bridgeport.edu

Abstract— In this work we present a dynamically controlled robot platform (Table-Top robot) to draw a portrait picture captured by a camera. This is done in three modules: image capture, image process, and robot control. This paper discusses the system’s architecture, methodology, and results. A Cartesian robot is used to do the work and the control program is done in MATLAB.

Keywords—X-SEL controller, IAI Table-Top robot; control system; image processing; dynamic control.

I. INTRODUCTION

Industrial robots are usually designed for specific applications. Most of the tasks in these applications are repetitive and dull for manufacturing processes. For this purpose predefined sequences and positions usually have to be stored in the robot’s built-in controller memory. Robots are often employed to do tasks such as welding, assembly, and packaging. These tasks are dull and boring, but robots nowadays are very autonomous and can perform tasks other than the typical industrial jobs. Most of these robots are equipped with a serial port to control the robot dynamically and remotely if needed. In this case the built-in controller will act as a slave and the remote computer will be the master. The movement of the robot will be based on the command string sent by the remote controller using the robot’s built-in controller’s language.

Art is a universal language and has been essential to human life for centuries. In this project we apply a typical industrial robot to produce picture portrait. This Table-Top robot is a Cartesian coordinate robot [1]. It has three prismatic joints whose axes are mutually orthogonal, as shown in Figure 1. This Table-Top robot is IAI made with a built-in X-SEL controller [2]. A pen is attached to the robot z-axis.

MATrix LABORatory (MATLAB) is a high-level language and interactive environment for numerical computation, visualization, and programming. MATLAB software is used in this paper for image processing, serial communication, and robot controlling.

The remainder of the paper is structured as follows: the subsequent section describes the methodology used for the system. Section III explains the image processing module. Section IV illustrates the serial connection between the PC and the robot, and the controller and communication program

algorithm. Section V presents the results. The final section concludes the paper.

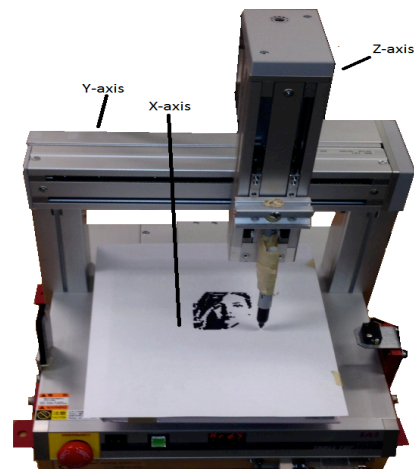


Fig. 1. IAI Table-Top robot.

II. METHODOLOGY

The working methodology of this robotic system is implemented in the below three modules :

- 1) Image capture
- 2) Image processing
- 3) Dynamic control of the robot

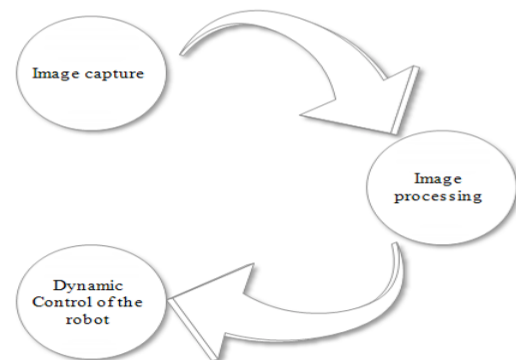


Fig. 2. Working methodology diagram.

III. IMAGE PROCESSING MODULE

The original image is captured with a camera and fed to the PC that controls the robot. The final image reproduced by the robot is of the dimensions 105x105 mm, where 2 pixels represent 1 mm. For this reason the original image has to be processed, converted to black and white, and finally resized as illustrated in Figure 3.

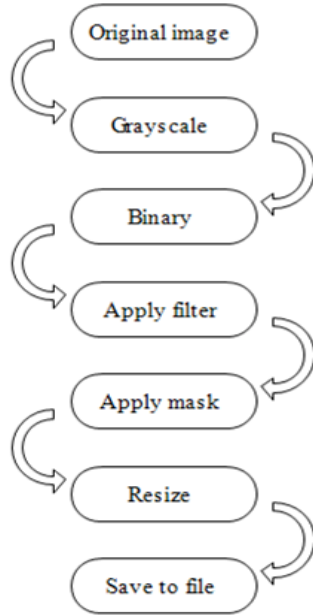


Fig. 3. Algorithm sequence of image processing module.

The image processing module follows the following steps:

- The original image file is read, as shown in Figure 4 (a).
- The original image is converted to a grayscale image, as shown in Figure 4 (b).
- The grayscale image is converted to a binary image based on a threshold, either by using the auto threshold or user defined threshold value. A threshold value of 0.5 is used for Figure 4 (c).
- An averaging filter is applied on the binary image using a 3-by-3 filter containing equal weights, as shown in Figure 4 (d).
- A sliding neighborhood mask is applied to set each pixel to the minimum value in its 2-by-2 neighborhood, as shown in Figure 4 (e).
- The processed image is rescaled to 150x150 pixels, as shown in Figure 4 (f).

The resulting image is saved to a file in the form of a binary matrix.

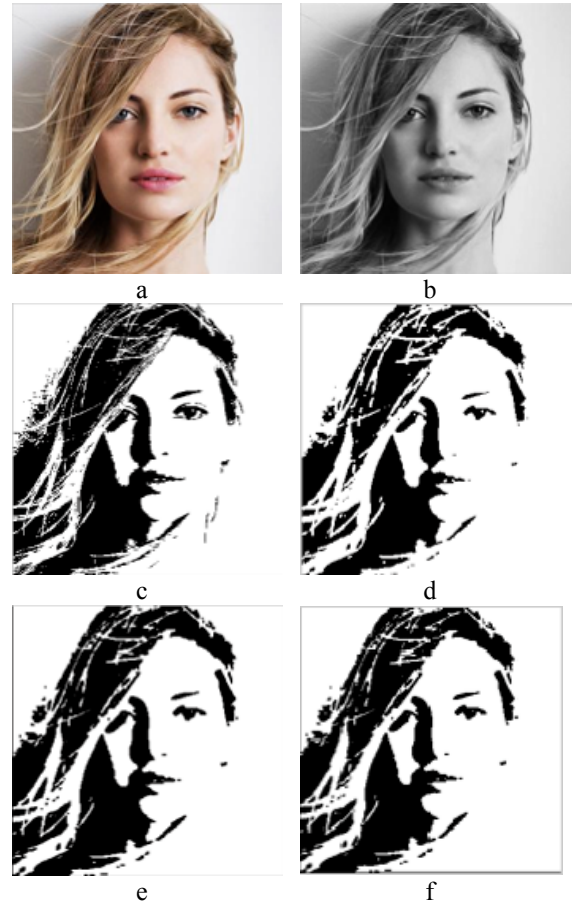
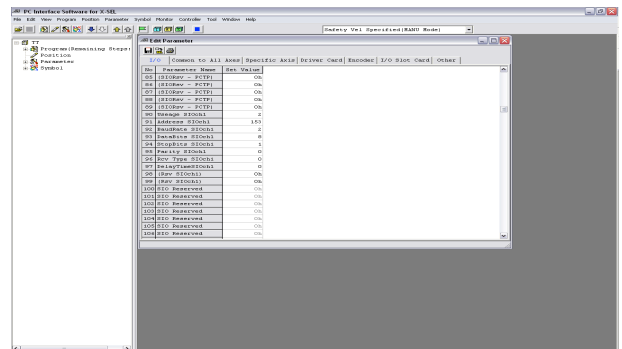


Fig. 4. Image processing module results: (a) original image, (b) grayscale image, (c) binary image, (d) filtered image, (e) masked image, (f) rescaled image.

IV. DYNAMIC CONTROL MODULE

A. Serial communication

To establish serial communication successfully between the PC and the robot's built-in controller, some of the I/O parameters on the robot's X-SEL controller need to be modified using the "PC interface software for X-SEL controller". Figure 5 shows a snapshot of the I/O parameter window on the software.



A USB cable is used to connect the PC to the robot. Table I lists the X-SEL controller remote communication parameters and their set values.

TABLE I. X-SEL COMMUNICATION PARAMETERS

Parameter No.	Value	Description
90	2	Select IAI Protocol B
91	153	Set the station number to 153
92	2	Set the baud rate to 38.4 kbps
93	8	Set the data length to 8
94	1	Set the stop bit length to 1
95	0	Set the parity type to None

The command sent from the PC has to follow the format shown in Figure 6. For example, if we want to move X and Y axes to coordinates 200 and 200 mm, the command string '!9923403001E001E012C00030D4000030D40@@@' will be sent, where :

- '!' is the header
- '99' is the station number
- '234' is the message id
- '03001E001E012C00030D4000030D40' is the content of the command.
- '@@@' means the checksum is disabled

For complete reference and X-SEL controller commands please refer to [3].

Header (start)	Station number	Message ID	Content of command	SC (checksum)	CR (End)	LF (End)
----------------	----------------	------------	--------------------	---------------	----------	----------

Fig. 6. X-SEL controller serial command format [3]

B. Controller program

The controller program is designed to work as an open-loop control system. In this system we do not request the current position from the robot. By doing so we reduced the system complexity and the required time to complete the process. In general the control program is designed to do the following:

- Establish communication with the robot.
- Turn on the axes' motors.
- Move the end-effector to home position.
- Load the image matrix file.
- Scan the matrix and draw the picture.

The scanning of the binary image matrix is done row by row. Based on the pixel value from the matrix, a command string is dynamically generated and sent to the robot. The pen is moved down on the paper for every black pixel and moved

up for every white pixel, unless the pen has been down or up previously. The flowchart in Figure 7 demonstrates the algorithm of the control program. And Table II explains the axes' arrangement.

TABLE II. AXES MOVEMENT ARRANGEMENT

Robot axis	Function	Target
X	Moving the paper forward and backward	Pictures' rows
Y	Moving the pen right and left	Pictures' columns
Z	Moving the pen up and down	Pen down and up

The robot speed is chosen to be coordinated with pauses in the control program. These pauses will allow the previous command to be executed before the next command is sent. Table III presents the moving characteristics of the commands.

TABLE III. ROBOT MOVING CHARACTERISTICS

Speed	Acceleration	Deceleration
500 mm/s	0.3G	0.3G

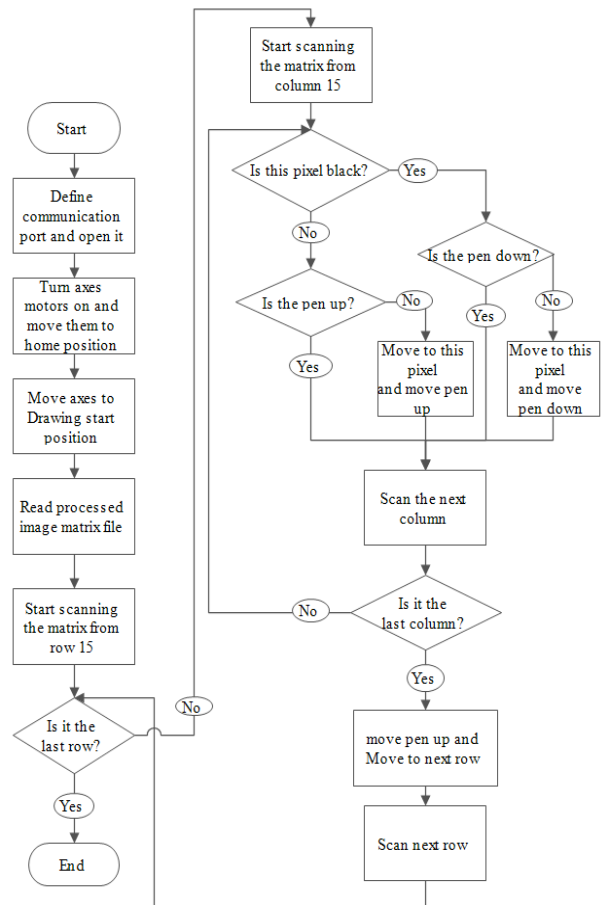


Fig. 7. Control program algorithm flowchart.

V. RESULTS

Figure 8 presents a comparison of the original images, processed images and the images drawn by the robot. The average time required to reproduce the images was around 7 minutes. Some factors that influence the image drawn by the robot are: the paper type, the pen type and the tip-size of the pen. For example, when we tried to use a marker with a large size tip we got a gap-free picture but we lost the fine details from it due to ink blotting. These details are important because the robot will transform a 135x135 pixels (35x35 mm) processed image into a 135x135 mm size image on the paper. And when the control program scans one pixel and leaves the next one, then the real size of the drawn image is approximately 65x65 pixels (18x18 mm). A video of the implementation and more project details can be found at <https://sites.google.com/site/alkhorobotic/home>

VI. CONCLUSION

In this paper we implemented a dynamically controlled robotic system to draw a portrait picture. The concept of the artist robot that we introduced in this paper is one of robots'

many interesting applications. Such applications will encourage and attract students and professionals alike to the field of science and engineering, and more specifically, robotics and automation.

ACKNOWLEDGMENT

We would like to thank Mr. Chris Prior for his support and providing IAI TT robot documents.

REFERENCES

- [1] Haider A. F. Almurib, Haidar Fadhil Al-Qrimli, Nandha Kumar, "A Review of Application Industrial Robotic Design," 2011 Ninth International Conference on ICT and Knowledge Engineering, 2012.
- [2] "Table-Top operational manual," IAI AMERICA Inc, Forth Edition, 2007.
- [3] "X-SEL Serial Communication Protocol Specification," IAI AMERICA Inc., Ninth Edition, 2010.

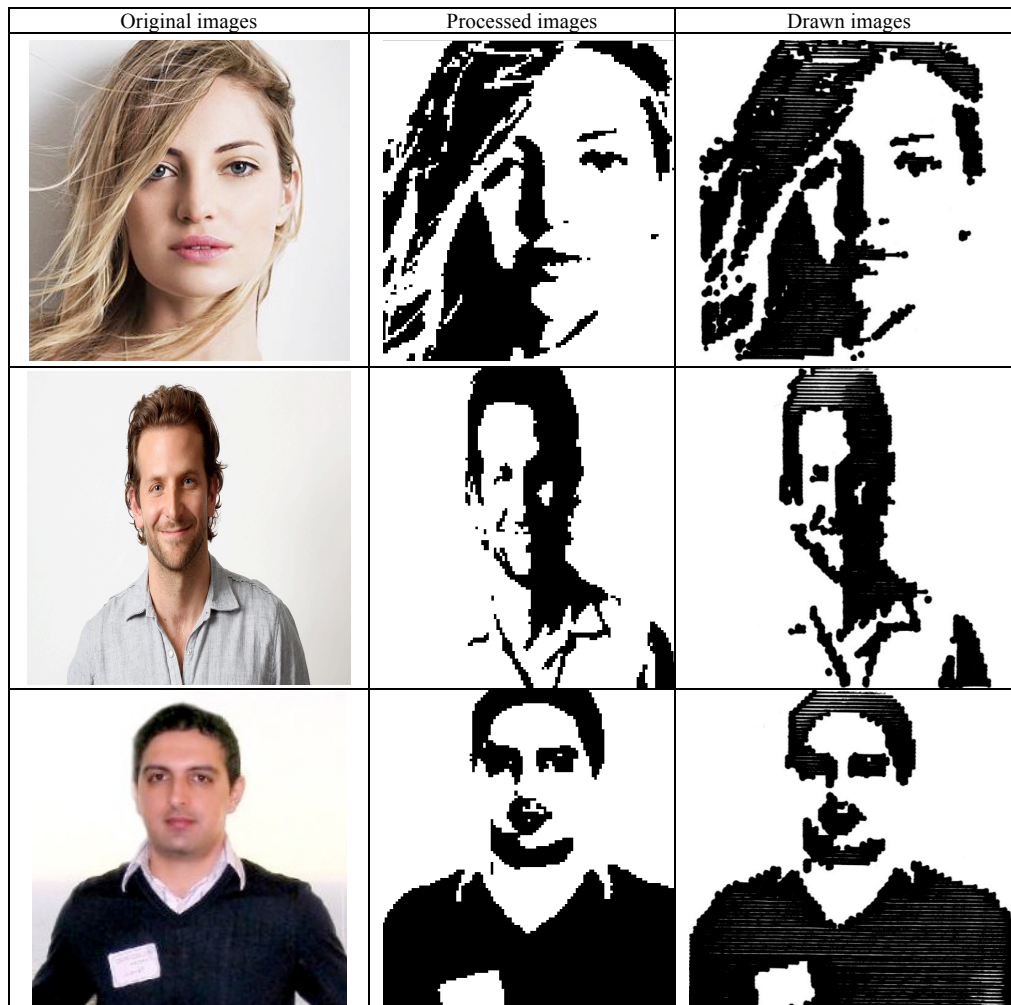


Fig. 8. Images drawn by the robot compared to the original and processed images.