

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 processing, and dynamic robot control. This poster discusses the system's architecture, methodology, and results. A cartesian robot (Table-Top Robot) is used to do the work and the control program is done in MATLAB.

Introduction

Industrial robots are usually designed for specific applications. Most of the tasks in these applications are repetitive and dull industrial processes. Robots are often employed to do tasks such as welding, assembly, and packaging. For this purpose predefined sequences and positions usually have to be stored in the robot's built-in 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 is based on a dynamically generated command string sent by the remote controller based on the robot's built-in controller's language. In this project we apply a typical industrial robot to produce picture portrait. It has three prismatic joints whose axes are mutually orthogonal, as shown in Figure 1. This Table-Top robot is an IAI made robot with a built-in X-SEL controller. A drawing pen is attached to the robot's z-axis.

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

Methodology

The working methodology of this autonomous robotic system is implemented in three modules as shown in figure 2.

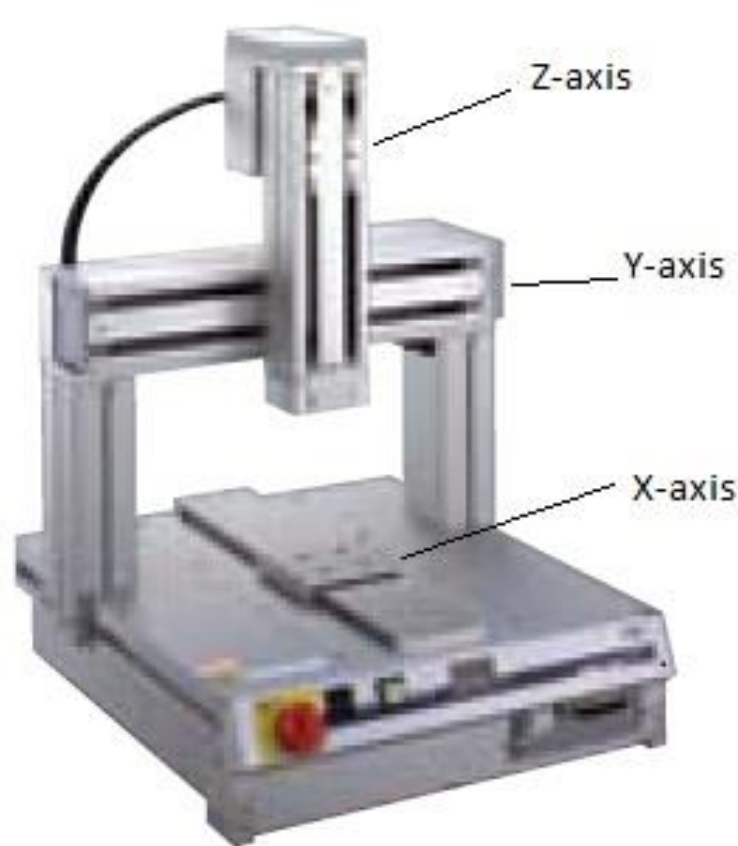


Figure 1: IAI Table-Top Robot.

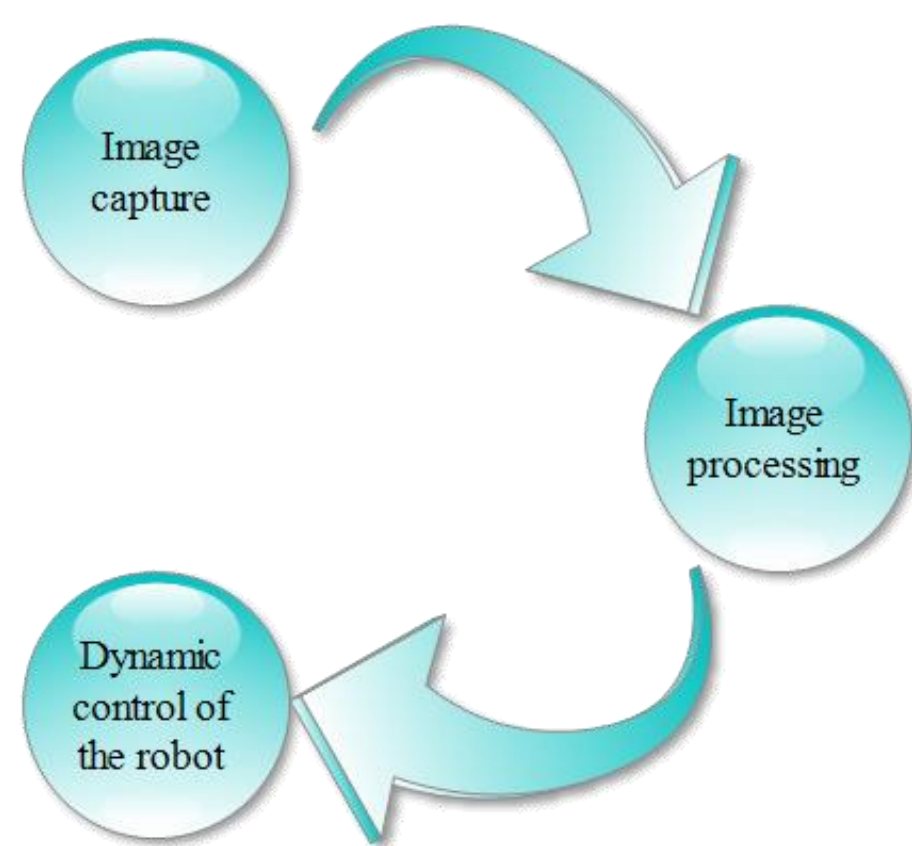
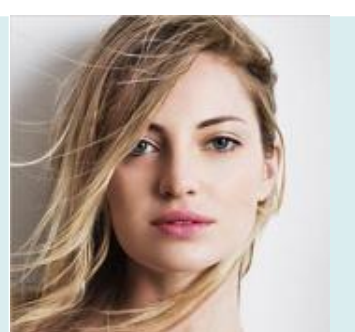


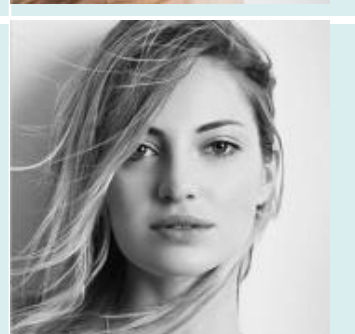
Figure 2: Working methodology diagram

Image Processing Module

The captured image or input image file.



The original image is converted to a grayscale image.



The grayscale image is converted to a binary image, by using a threshold value of 0.5.



An averaging filter is applied on the binary image using a 3-by-3 filter containing equal weights.



A sliding neighborhood mask is applied to set each pixel to the minimum value in its 2-by-2 neighborhood.



The processed image is rescaled to 150x150 pixels, and saved to a file in the form of a binary matrix.

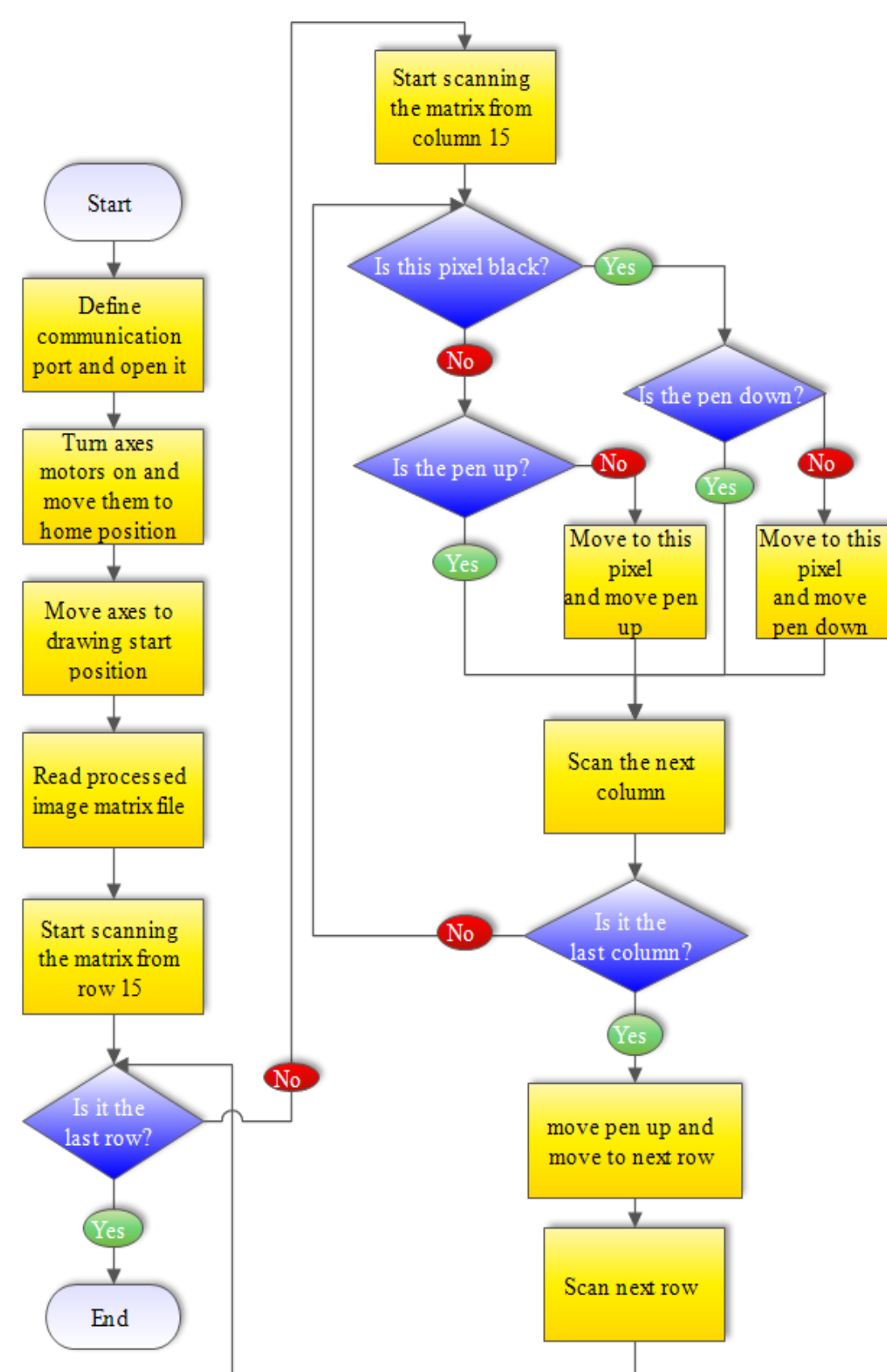


Figure 4: Control program flowchart.

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.

Dynamic Control Module

a) Serial Communication Module

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". Table 1 lists the X-SEL controller remote communication parameters and their set values. A USB cable is used to connect the PC to the robot.

Table 1: 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 5. For example, if we want to move X and Y axes to coordinates 200 and 200 mm, the command string 'I9924500100103001E001E01F400030D4000030D40 @@' will be sent. The X-SEL controller commands are described in "X-SEL Serial Communication Protocol Specification".

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

Figure 5: X-SEL controller I/O parameter window.

b) Control Program Module

The control 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:

- 1) Establish communication with the robot.
- 2) Turn on the axes' motors.
- 3) Move the end-effector to home position.
- 4) Load the image matrix file.
- 5) 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 4 demonstrates the algorithm of the control program.

The robot speed is chosen to be coordinated with pauses in the control program. These pauses will allow enough time for the previous command to be executed before the next command is sent.

Results

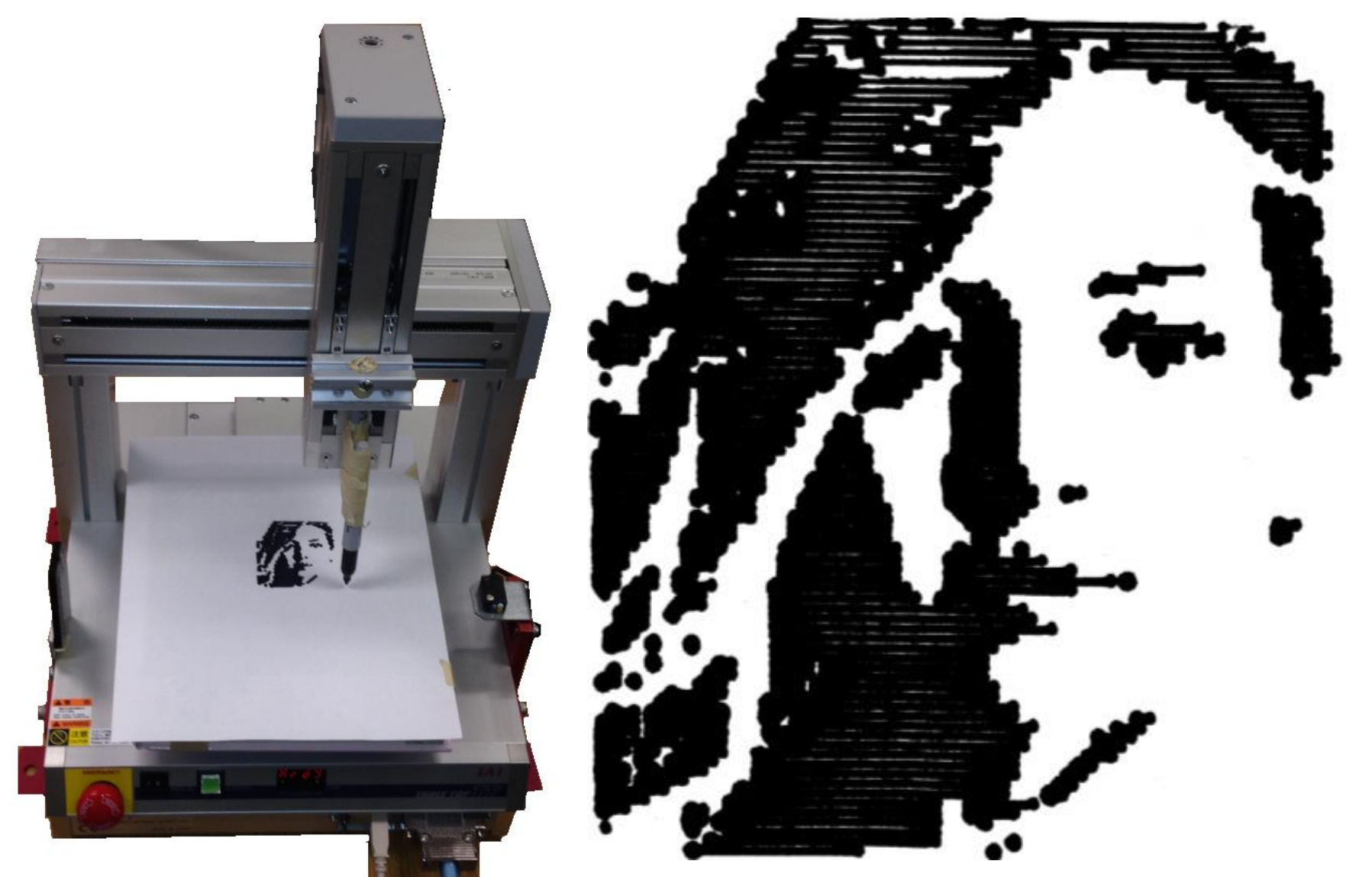


Figure 6: Image drawn by the TT-robot.

Figure 6 presents the final images drawn by the robot. The average time required to draw the picture was around 7 minutes. Some factors that influence the picture 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).

Conclusion

In this project we implemented a dynamically controlled robotic system to draw a portrait picture. The concept of the artist robot that we introduced in this poster 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. A video of the implementation and more project details can be found at <https://sites.google.com/site/alkhorbotic/home>