Michelangelo: A 3-DOF Sketching Robot

Ravi Sankar Ippili, Uttam Grandhi, Dr. B.K. Raut, Dr. R.K. Mittal
Mechanical Engineering Group
Birla Institute of Technology and Science (BITS) -Pilani
Pilani, Rajasthan, INDIA
ravi.sankar.ippili@gmail.com, uttam.grandhi@gmail.com, rout@bits-pilani.ac.in, rkm@bits-pilani.ac.in

Abstract— **This paper presents an indigenously developed artbot capable of drawing complex images. The innovativeness lies in the capacity to draw images with multiple gray values. The robot possesses a tool magazine that holds pencils of varying shades of gray. The controller chooses the type of pencil based on the intensity levels of each gray pixel. Geometric transformations like scaling and rotation can be performed on the image before drawing. To enhance the characteristics of the image available image processing algorithms are used. A GUI is developed with minimal buttons to enable easy interaction with the robot. The simple yet efficient software and hardware design makes it a unique sketching robot.**

I. Introduction

Art surrounds us everywhere. It has become so inevitable that even technical fields were also influenced by it. Robotics is a field that clearly signifies such an effect. Growing interest to keep robots as sources of entertainment resulted in a new field of robotics, 'artbots'. Robots that can imitate humans in the true sense of an artist have always been a fascination to all age groups.

During the last decade, various models of sketching robots have been developed that can draw pictures. They function similar to a printer that prints sequentially line by line and hence, lack the dexterity of a human artist. These models also fail to emulate a scene accurately as a fixed end effector, usually a pencil or a pen, is used to represent various gray levels in an image. Sketching Robots that have been developed recently lack the portability and simplicity [1], [5], [6]. These robots sketch portraits only with a single color. This motivated us to develop a robot which is compact in size, user friendly and a source of entertainment. We have also incorporated multi shades of gray in our model. Further, the graffiti output device [4] that has been developed for sketching involves the use of ink, which has an inconvenience of drying the portrait once it is drawn. This suggested us to use pencils and thus enabling the user to access the drawn figure immediately.

In this paper, we present the design of a 3-DOF sketching robot that sketches the picture by using a novel contour extraction algorithm and a tool magazine that can hold pencils of varying darkness to represent multiple gray levels in the image. The behavior of the robot resembles that of an artist, who would, first, analyze the object, and then, draw the image on a piece of paper. The application runs in real-time and is interactive. It uses existing tools for image processing, robot control and object manipulation. Sketching complex objects requires human-specific skills, and is thus a challenge for robots.

The rest of the paper is organized as follows: Section II presents the system set-up and the tools used. Section III presents the strategy used to realize human-like drawing of images. Section IV discusses the image acquisition, image conversion, image preprocessing, threshold determination, image binarization and inverse kinematics. Section V discusses the results and shows the possible extensions of the system.

II. Experimental Set-Up

A. Overview

The 3-DOF model of the sketching robot is made with the use of 4 servo actuators (ROBOTIS AX-12). 2-DOF are used to sweep the area on the paper. The other 1-DOF is used to implement pen-up and pen-down states. The actuators have torque-bearing capacity of 12 kgf.cm at 7 V and 16.5 kgf.cm at 10 V. The holding brackets were modeled using *PRO-E* and were manufactured with fused deposition modeling machine. The algorithms extract contours from the image using some edge detection techniques developed by Canny [3]. The end-effector is equipped with a magazine with a maximum capacity of 4 pencils to portray various gray level values in the image. The actuators are connected in a daisy chain mechanism and interfaced through a USB port with the computer. The Graphical User Interface was developed using *Visual Basic*. The instruction packets were constructed by the controller and sent to the actuators for execution.

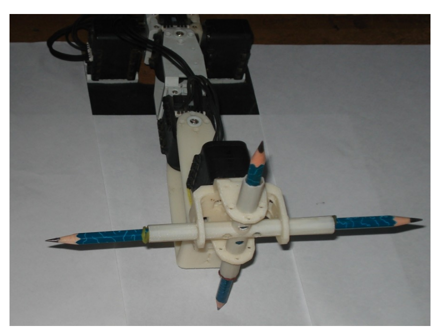

Figure 1: Photograph of the Michelangelo developed showing the magazine with four pencils

III. STRATEGY

The sketcher traces the contour of a given image by joining all the scattered dots with pen-up and pen-down states. Depending on the desired range of the sketcher a "Reachable Work Space (RWS)" has been fixed. RWS is the area between two semi circles obtained by the extreme configurations of the sketcher. In the arm, 1-DOF is used for pen up and pen down, while the other DOF are used to traverse a plane. The robot is controlled by an algorithm developed that optimizes the total distance traveled by the end effector, thus accomplishing the task in minimum time. The image to be processed is loaded through a GUI and converted to a desired format using the inbuilt functions of MATLAB. Inverse Kinematics is implemented for the 2-DOF planar arm to find the joint parameters. Since a 2-DOF manipulator can reach any point on a plane in two different arm orientations, a procedure has been devised to choose suitable and feasible orientation. Using the positions and speeds of all the actuators instruction packets are constructed by the software. These instructions packets are sent to the actuators connected in daisy chain through USB interface, to draw the image

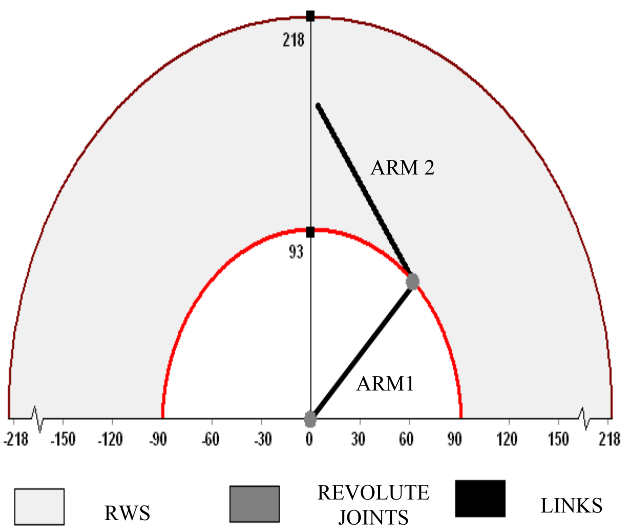

Figure 2: Reachable Work Space of Michelangelo

IV. DATA PROCESSING

A. *Image Acquisition*

This is the first and the foremost step which forms the basis for the remaining phases of data processing. We have used a CCD based image sensor for image acquisition. The reason we have chosen CCD sensors over CMOS sensors is their ability to create high quality, low noise images. The picture of desired object is taken by placing it in front of the image sensor. The lighting conditions are maintained same for all the pictures taken to provide similar ambience in the scene.

B. *Image Conversion*

The image acquired in previous step is transformed to a desired format to perform various operations on it. The image is initially in JPEG format. It is converted to PGM (Portable Gray Map) format using image processing tool box inbuilt in *MATLAB*. The inbuilt function used to achieve this is *rgb2gray*. The function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. For each pixel in the image the luminance value $L \in \{0,1,...255\}$. The luminance values 0 and 255 correspond to black and white colors respectively.

C. *Image Preprocessing*

The digital image acquired is prone to a variety of types of noise. There are several ways that noise can be introduced into an image, depending on how the image is created. We have implemented some of the noise reduction algorithms provided in the Image processing toolbox.

One of the most common types of noise, Salt and Pepper noise, typically seen on images, represents itself as randomly occurring white and black pixels. An effective noise reduction method for this type of noise involves the use of median filter. Median filter is applied using *medfilt2* function of the toolbox. The function performs two dimensional median filtering. Median filtering, unlike linear filtering, besides removing noise it also preserves edges of the image, thus manifesting itself as an effective image enhancement filter.

D. *Determination of Threshold*

The threshold for the processed image is determined using gray level histograms and inbuilt functions of MATLAB. The inbuilt function *graythresh* computes the global threshold of the image using Otsu's method [2]. The image retains all the important characteristics and has an esthetic look. Besides this technique, various gray level histograms were plotted to determine the threshold based on the frequency scatter of the gray values.

E. *Image Binarization*

For each pixel in the image, the luminance $L \in \{0,1,....255\}$ is transformed to either 0 or 1. The algorithm converts the image to a binary representation that can be drawn by the robot using pencils on white paper. The threshold determined in the previous step is used to achieve this.

F. *Inverse Kinematics*

Inverse kinematics algorithms were run to compute joint angle parameters from the Cartesian coordinates of the 2 DOF planar RR arm configuration. Every point (x^*, y^*) on the RWS can be reached by the end effector in two different arm orientations. Taking into consideration various mechanical constraints associated with the manipulator, a particular arm orientation was chosen. Some of the important equations used in the development of inverse kinematics module are as follows:

$$r = \sqrt{(x^*)^2 + (y^*)^2} \tag{1}$$

$$\theta_2 = \cos^{-1}\left(\frac{l_1^2 - r^2 + l_2^2}{2 l_1 l_2}\right) - \pi \tag{2}$$

$$\theta_1 = \sin^{-1}\left(\frac{x^*}{r}\right) + \cos^{-1}\left(\frac{l_1^2 + r^2 - l_2^2}{2 l_1 r}\right) \tag{3}$$

Where,

x^*, y^* are the abscissa and the ordinate of a point in RWS,
r is the absolute distance of (x^*, y^*) from the origin,
l_1, l_2 are the link parameters of the manipulator, and
θ_1, θ_2 are the joint parameters of the manipulator.

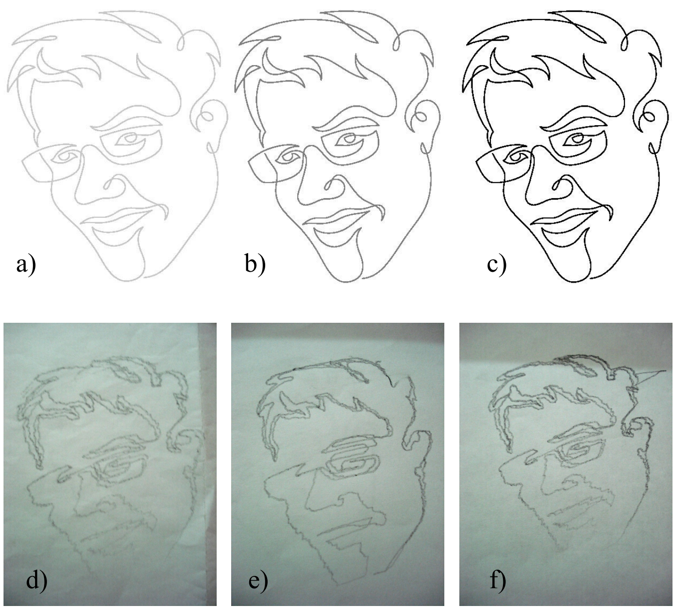

Figure 3: Human Face. Images drawn with decreasing gray scale values. a), b), c) correspond to gray levels 200, 128 and 0 respectively. d), e), f) are the corresponding images drawn with HB, 4B and 6B pencils.

V. RESULTS, DISCUSSION AND FURTHER WORK

Some results are presented in Fig. 3 and 4, which are scan of sketches drawn by Michelangelo. Sketching continuous line drawing images (Fig 3.) is not an easy task, even for humans. Here, the robot performs this task skillfully. By planning a smooth trajectory, controlling the speed and acceleration of the arm, the manipulator deftly sketches the image.

As we see from the test figures (Fig. 4.), lines are not smooth and straight. It is mainly due to errors in the model of the robot used during the inverse-kinematics process. More specifically, the position of the pencil is supposed to be fixed, but this can also slightly change during the drawing. Errors are also introduced in the set-up (e.g. if the links are not properly clamped.).

Each step of the whole process can be monitored and controlled visually, and parameters can be changed online, if desired. However, the application is created to run without the intervention of the user. The whole process is fast: the image processing part and inverse kinematics take only a few seconds, and the drawing takes less than 10 minutes. A similar portrait sketched by a human would clearly take more time to draw.

In further work, we plan to add more interactive behaviors to the robot. E.g. the robot may ask the user whether he/she would like to have some more personalized addition be made to the drawing. We are in the process of incorporating stereoscopic vision system, which could be used to implement robust object tracking.

The process of transforming the image in a *black & white* representation could be improved in many ways. For instance, in thresholding process, other criteria such as a *solidity* measure could be considered, to better distinguish the object features. *Solidity* is a measure of a surface divided by the convex surface surrounding the group of pixels. It could, then, be possible to detect if multiple features are merged together, resulting in non-convex areas.

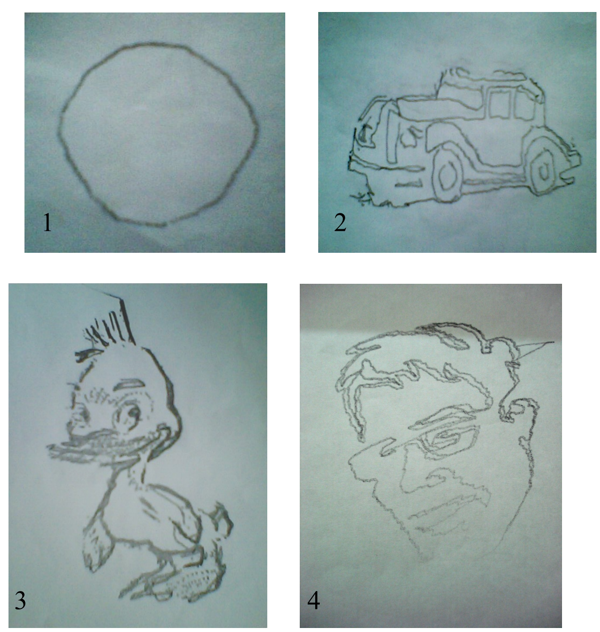

Figure 4: Test Figures. Images drawn by the robot range from simple geometries (*1*) to medium (*2) and* complex figures(*3 &,4)*.

TABLE I. COMPARISION BETWEEN DIFFERENT TEST FIGURES

Image	Image Complexity	Time Taken [min.]
Circle	Low	0.66
Car	Medium	4
Human face	Medium	6
Duck	High	7

VI. CONCLUSION

This paper presented a simple realization of a robot capable of sketching complex objects. We have used novel methods to realize a sketch with multiple gray shades and developed a lucid user interface to reduce unwanted user interference. The contribution of this work to art robots is two-fold: first, it demonstrates how image processing techniques and classical inverse kinematics algorithms can be tuned to endow an artbot with human-like competencies; second, it highlights the potential of sketching robots as source of entertainment.

REFERENCES

[1] Sylvain Calinon, Julien Epiney and Aude Billard, “A humanoid robot drawing human portraits", EPFL, Switzerland

[2] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66

[3] J. Canny, “A computational approach to edge detection,” *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[4] J. Lehni, “Hektor: a graffiti output device,” 2002, diploma project at the Ecole Cantonale d’Art de Lausanne. Available at http://www.hektor.ch.

[5] M. Gommel, M. Haitz, and J. Zappe, “Robotlab autoportrait project: Human portraits drawn by a robot,” 2004, center for Art and Media Karlsruhe. Available at http://www.robotlab.de.

[6] M. Stein and C. Madden, “The pumapaint project: Long term usage trends and the move to three dimensions,” in *Proceedings of the IEEE Intl Conference on Robotics and Automation (ICRA)*, Barcelona, Spain,April 18-22 200