International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 4, Issue 2, pp: (112-120), Month: April - June 2016, Available at: www.researchpublish.com

The Electronic Paintbrush

[1]Rohini Jayachandre G., [2]Swarna Raghavi S., [3]Prof. Vidhyapathi CM.

School of Electronics Engineering, VIT University, Vellore, Tamil Nadu, India

Abstract: **The electronic paint brush in this project is designed to sketch regular shapes like a circle, square and a rectangle. The central idea incriminated in this project is to detect objects with regular shapes in an image using image processing in MATLAB®, thereby transfer the information of object to PIC16F using UART. This will control the robot to sketch the detected object in MATLAB® using image processing. The robot is programmed to move in particular directions with respect to the instruction received at serial communication port of microcontroller.**

Keywords: **Image processing, UART (Universal Asynchronous Receiver/ Transmitter, PIC (Peripheral Interface Control), MATLAB® Software.**

1. INTRODUCTION

When a conventional brush is used to paint, it cannot make perfect edges as there are high chances that the bristles are wider than required when it comes to edges. Also when work is done manually fine work is not assured. Hence this project prioritizes on making accurate movement of the paint brush (or pencil or pen) fixed to the robotic arm. Secondly, this kind of electronic paint brush is not restricted to the smooth surfaces. The robot can be programmed to sketch the image on any surface. As we see there are advertisements painted on a 10+ storied (high rise) buildings, where artists find hard time to concentrate at the edges and they make use of stencils involving greater risk and spending lot of time. In similar instances symmetry is also of concern, where one cannot draw exact figures on either side. Instead they can make use of an electronic paintbrush which can sketch their design any number of times in less time with zero risk factor. We are building this prototype for painting environment. But otherwise it can be used in industries where high precision and accuracy are of utmost importance. Instead of a paintbrush if we can fix a drilling tool, it can cut ply board in required shape. Similarly various applications can arise.

The primary aim of the project is to design, develop and implement the Electronic Paintbrush which helps to achieve a painting device. We designed an electronic paintbrush which can draw simple images. The design of this brush comprises of a microcontroller, a robotic arm and image processing through MATLAB®. The image of a circle or square is selected through MATLAB® and is recognized as a circle or square respectively and the signal is sent to the microcontroller through the serial communication port. The ARM is programmed to move in any particular direction with respect to the instruction received at serial communication port of microcontroller.

2. BACKGROUND

In the past there has been an extensive research contribution to electronic paint brush. Few of the discoveries include automatic wall painting robot [3], electronic paintbrush with scanners [1] and dispensers and hand held electronic paintbrush [2].

An automatic wall painting robot was made in the year of 2013, which was a four wheeled robot that is made capable of painting wall by spraying mechanism. The use of this is focused on interior wall painting. This robot is simple and portable. The robot is designed using few steels, conveyor shaft, spray gun and a controller unit to control the entire operation of the robot. This robot is compact because of high speed and pressure capabilities they have. They also have a very small weight to power output ratio and predictable performance *i.e.*, losses are minimum due to less number of moving parts and so gives expected performance. The block diagram (Fig. 1) is as shown below.

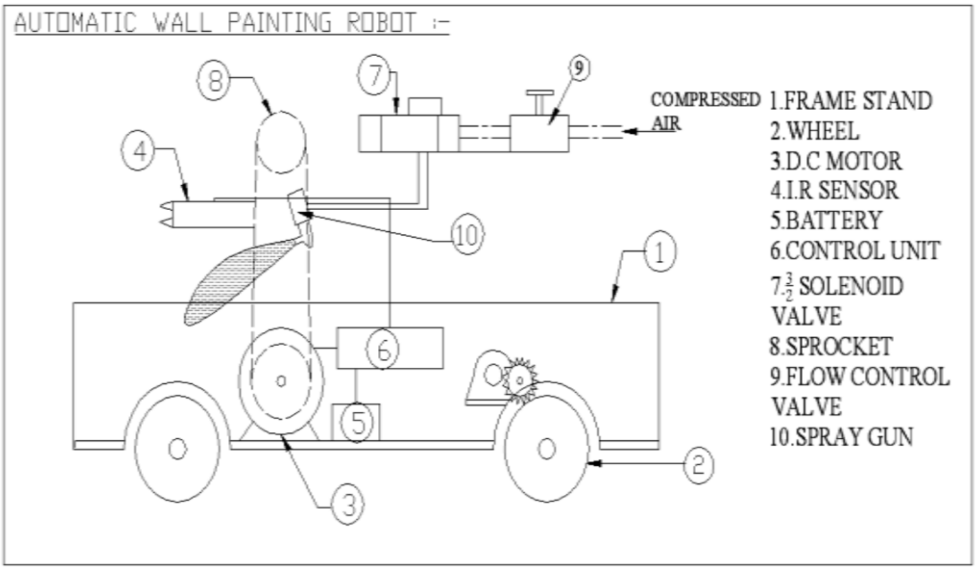

Fig. 1 – Block diagram of Automatic Wall Painting Robot

Electronic paint brush with scanners and dispensers was worked upon in the year of 2005. It is an electronic brush for dispensing ink onto a writable medium employs electronic-brush housing, at least one ink dispenser coupled to the electronic-brush housing, an electronic-brush scanner coupled to the electronic-brush housing, and a controller. The controller is in electrical communication with the ink dispenser and the electronic-brush scanner. A position of the electronic brush is determined based on at least one position indicator in a first portion of a dispensed image that is scanned by the electronic brush scanner and communicated to the controller. An ink dispense signal is sent from the controller to the ink dispenser based on the determined electronic-brush position. The block diagram (Fig. 2) is as shown below:

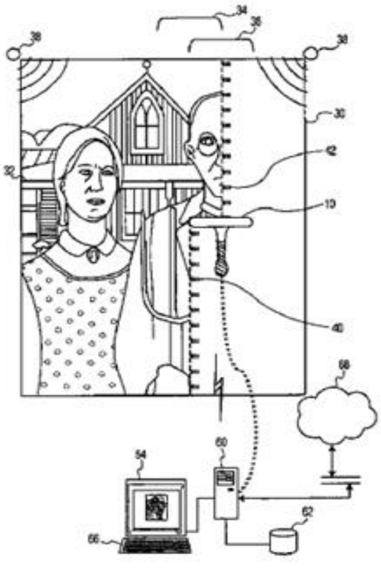

Fig. 2 – Block diagram of Electronic Paintbrush with Scanners and Dispensers

The above mentioned are certain kinds of paint brushes. Some have focused on painting large area and have eliminated the repetitive and cumbersome work required while some have focused on painting an intricate design on wall using dispenser and scanners. Most of the discoveries from the past have been restricted to smooth surface painting. The uniqueness of our project arises with the design of our two wheeled robot with three motors. Two motors to control the movement of wheels and another at the top to control the movement of brush. The dc motor at the top can enable us with painting on uneven surface as well. We have taken accuracy and dimensions into account. Hence when accuracy and symmetry is of concern repetitively this electronic paint brush of ours can be of immense help. We look forward to improve the product.

3. OUTLINE OF THE PROCESS

To get the required image and perform image processing using gray scale techniques and edge detection techniques in order to achieve object detection using the MATLAB® code which results in detecting regular shapes. In order to give output of the software as input to hardware we interface the two system components through hyperterminal software and Code the microcontroller PIC16F using mplab software, in order to control the movement of two stepper motors of robot.

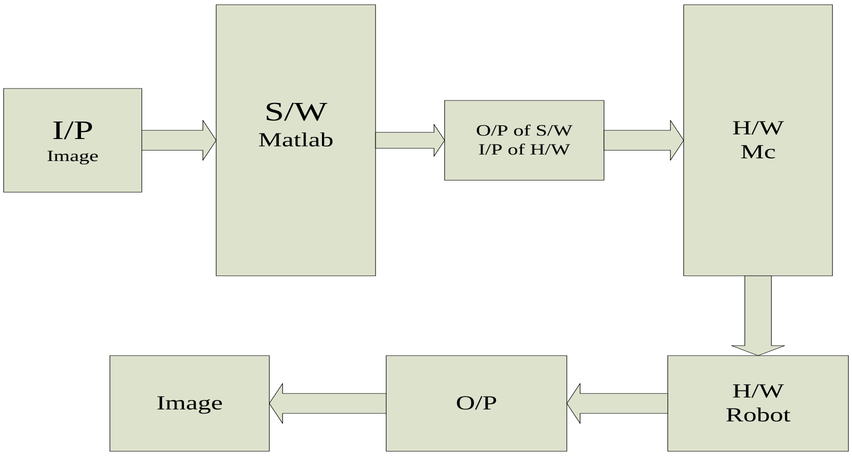

Fig. 3 Block Diagram

- Finally test the working model in different environments.
- The image loaded is converted to a binary image, which leaves black and white spaces according to gray scale and threshold value. Then find centroid of the required space using respective formulae.
- The objects in an image are thus detected according to the centroid location.

4. OBJECT DETECTION

RGB color model:

The **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a device-dependent color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same color across devices without some kind of color management.

To form a color with RGB, three coloured light beams (one red, one green, and one blue) must be superimposed (for example by emission from a black screen, or by reflection from a white screen). Each of the three beams is called a *component* of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture.

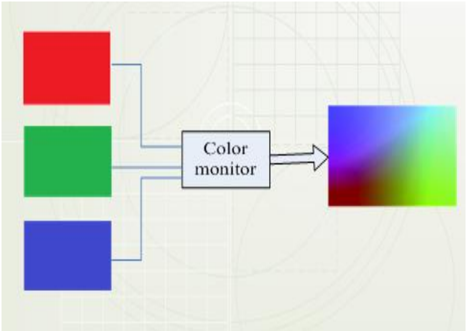

Fig. 4 RGB model

Gray scale image:

In photography and computing, a **grayscale** or **greyscale** digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bi-level or binary images). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the presence of only one (mono) color (chrome).

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, *etc.*), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image.

Image acquisition:

The first stage of any vision system is the Image acquisition stage. After the image has been obtained, various methods of processing can be applied to the image to perform the many different vision tasks required today. However, if the image has not been acquired satisfactorily then the intended tasks may not be achievable, **even** with the aid of some form of image enhancement.

Image Acquisition Toolbox™ enables you to acquire images and video from cameras and frame grabbers directly into MATLAB® and Simulink®. You can detect hardware automatically and configure hardware properties. Advanced workflows let you trigger acquisition while processing in-the-loop, perform background acquisition, and synchronize sampling across several multimodal devices. With support for multiple hardware vendors and industry standards, you can use imaging devices ranging from inexpensive Web cameras to high-end scientific and industrial devices that meet low-light, high-speed, and other challenging specifications.

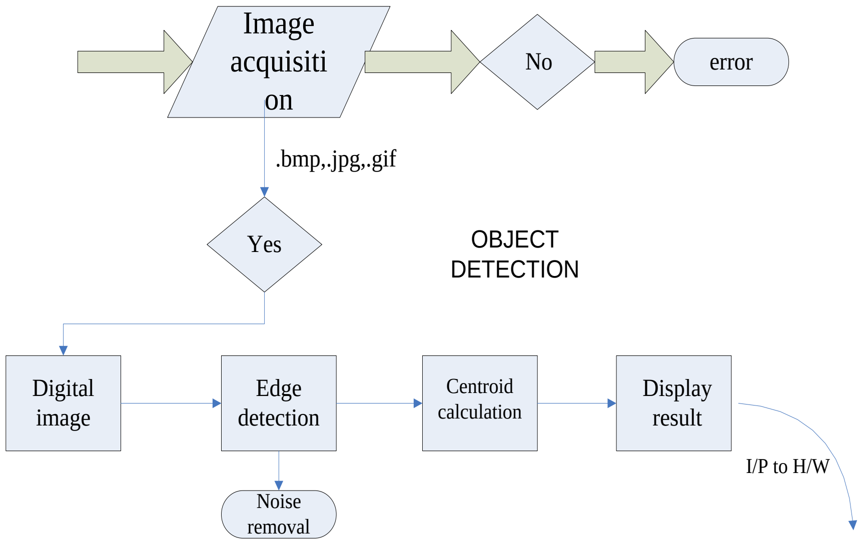

Fig. 5 – Flow chart of Object Detection

Threshold

The simplest approach to segment an image is using thresholding.

If $f(x, y) > T$ then $f(x, y) = 0$ else $f(x, y) = 255$

One of the thresholding methods is illustrated as follows:

Automatic thresholding:

- To make segmentation more robust, the threshold should be automatically selected by the system.
- Knowledge about the objects, the application, the environment should be used to choose the threshold automatically:

* Intensity characteristics of the objects
* Sizes of the objects and number of different types of objects appearing in an image
* Fractions of an image occupied by the objects

RGB to gray scale conversion:

In grayscale images, however, we do not differentiate how much we emit of the different colors, we emit the same amount in each channel. What we can differentiate is the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels.

When converting an RGB image to grayscale, we have to take the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One such approach is to take the average of the contribution from each channel: (R+B+C)/3. A different approach is to let the weights in our averaging be dependent on the actual image that we want to convert, *i.e*., be adaptive.

Detection Algorithm:

The steps of the detection algorithm are now described and qualitatively. First the edges are detected (step 1) then the boundary fragments of the weak detectors, that form the strong detector, are matched to this edge image (step 2). In order to detect (one or more) instances of the object (instead of classifying the whole image) each weak detector hi votes with a weight whi in a Hough voting space (step 3). Votes are then accumulated in a circular search window (W(xn)) with radius dc around candidate points xn. The Mean-Shift modes that are above a threshold that are taken as detections of object instances (candidate points). The confidence in detections at these candidate points Xn is calculated using probabilistic scoring. The segmentation is obtained by back projection of the boundary fragments (step 3) of weak detectors which contributed to that centre to a binary pixel map. Typically, the contour of the object is over-represented by these fragments. We obtain a closed contour of the object, and additional, spurious contours (seen in figure 8, step 3). Short segments (< 30 pixels) are deleted, the contour is filled (using Matlab's 'filled area' in regionprops), and the final segmentation matte is obtained by a morphological opening, which removes thin structures (votes from outliers that are connected to the object). Finally, each of the objects obtained by this procedure is represented by its bounding box.

5. SKETCHNG THE DETECTED OBJECT

Serial communication:

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

A universal asynchronous receiver/transmitter is a computer hardware device that translates data between parallel and serial forms. The universal designation indicates that the data format and transmission speeds are configurable. The electric signalling levels and methods are handled by a driver circuit external to the UART. The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires.

Two Wheeled Robot:

Two wheeled robots are more energy efficient, tend to have a simpler mechanical structure, as well as simpler dynamics compared to that required by legged robots to make contact with the ground and provide a driving force. These robots have two coaxial wheels mounted on either side of an intermediate body, with a centre of mass above the wheel axles, and therefore, must actively stabilize themselves to prevent toppling.

Two-wheeled robots have a number of advantages over other mobile robots. Although they are more difficult to control than statically stable wheeled robots, two-wheeled robots are still much easier to control than legged robots. This wheel configuration makes them highly manoeuvrable, because of their ability to turn on the spot, similar to differential drive robots.

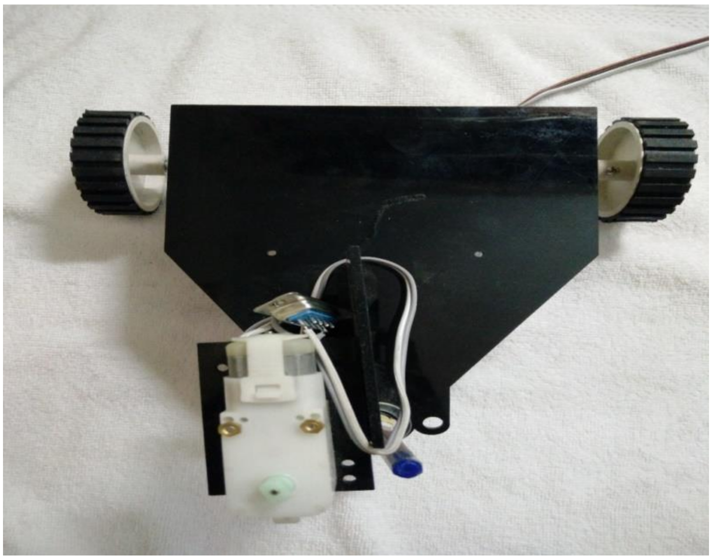

Fig. 6 – Robot -The Electronic Paintbrush

There are essentially three ways in which a user can communicate with the computer *via* MATLAB: through the Command Window, through the use of scripts and functions, and through GUIs. The Command Window is MATLAB's default I/O technique. As the name implies, it is a window into which any standard MATLAB commands or user-defined commands, such as "2+2" or "answer = conv2 (mask, filter)," can be typed. Although the Command Window is adequate for accomplishing simple tasks, it is often useful to create a file containing a list of such commands. These files are called scripts if they simply run a list of commands and functions if they accept input arguments and/or return output arguments. Both scripts and functions can be executed either directly from the Command Window or from within other scripts or functions. The third communication device, a GUI [4], (Graphical User Interface) provides an intuitive interface between the user and the programming language. A GUI allows the user to bypass MATLAB commands altogether and, instead, to execute programming routines with a simple mouse click or key press as shown in Fig.7 below.

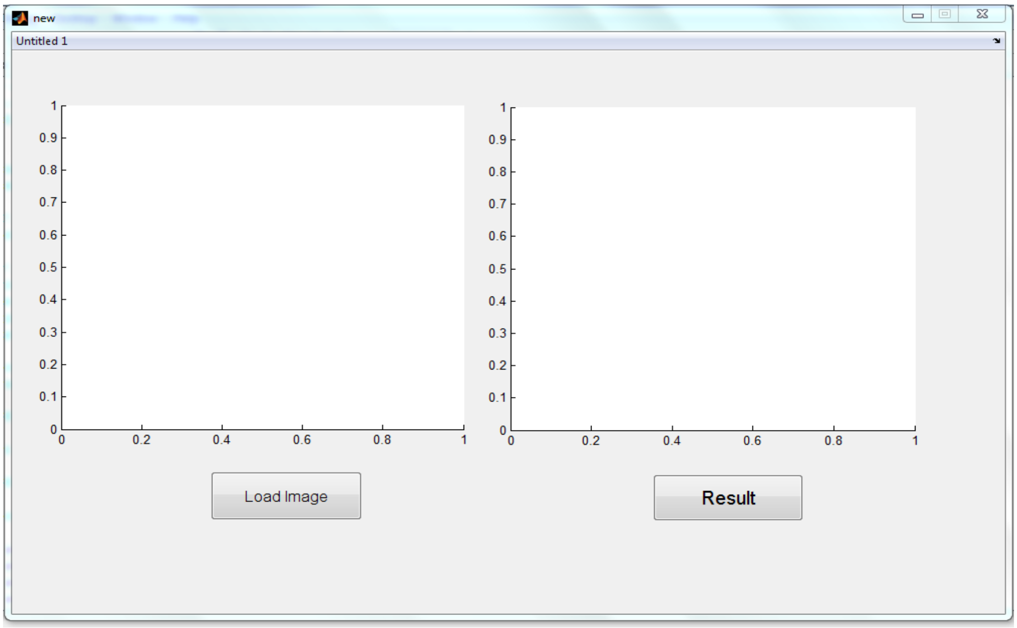

Fig. 7 GUI

This project does not necessitate the user to be acquainted with knowledge of MATLAB® or computer programming to successfully navigate a well-designed GUI; indeed, from the user's perspective, the language underlying the GUI is irrelevant. GUIs can range from simple question boxes prompting the user for a Yes/No response, to more complex interfaces, an example of which will be provided below. MATLAB provides the user with intuitive tools for easy construction of GUIs.

Hence Graphical User Interface (GUI) is used to implement the object detection section of project. Two graphs and two press buttons are used from the set up. These components will have their callback functions thus enabling us define the purpose of each component by manipulating the code.

The press button named "load image" is used so that it facilitates the user to choose an image of their choice for further processing. The press button named "result" will display the result after performing image processing techniques involved. The process is depicted with the following images and description.

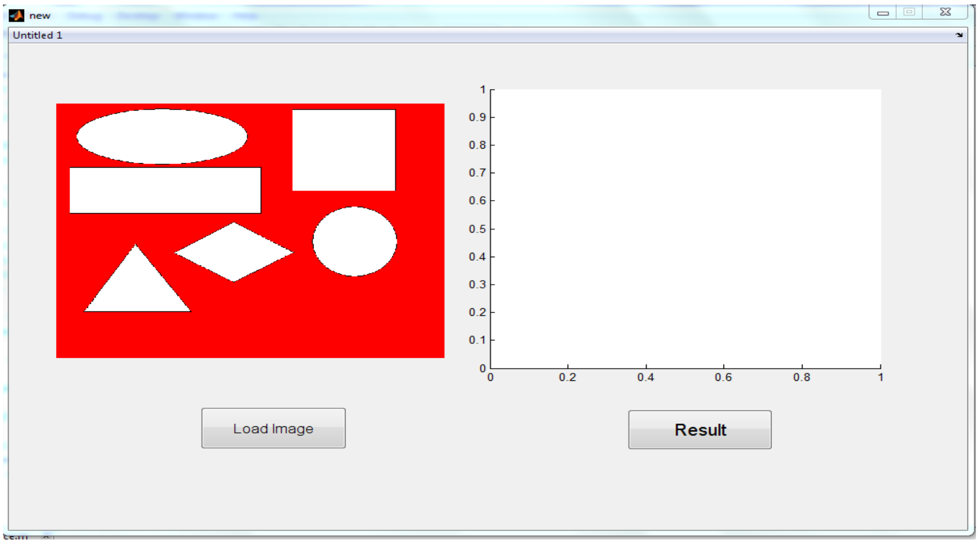

Fig. 8 Load image

In the above image (Fig. 8) the user is enabled to select an image of his/her choice. As we can see the user's choice of image is displayed in first graphic space. The image is comprised of different shapes which are not detected yet.

In preceding step when the user prefers to view result, the push button coined "result" should be pressed. This will enable the code to perform gray scale technique and perform object detection through edge detection technique. The result of object detection is obtained as depicted in figure 9 and in figure 10. As it can be seen multiple objects are detected in figure 9 and a single object is detected in figure 10.

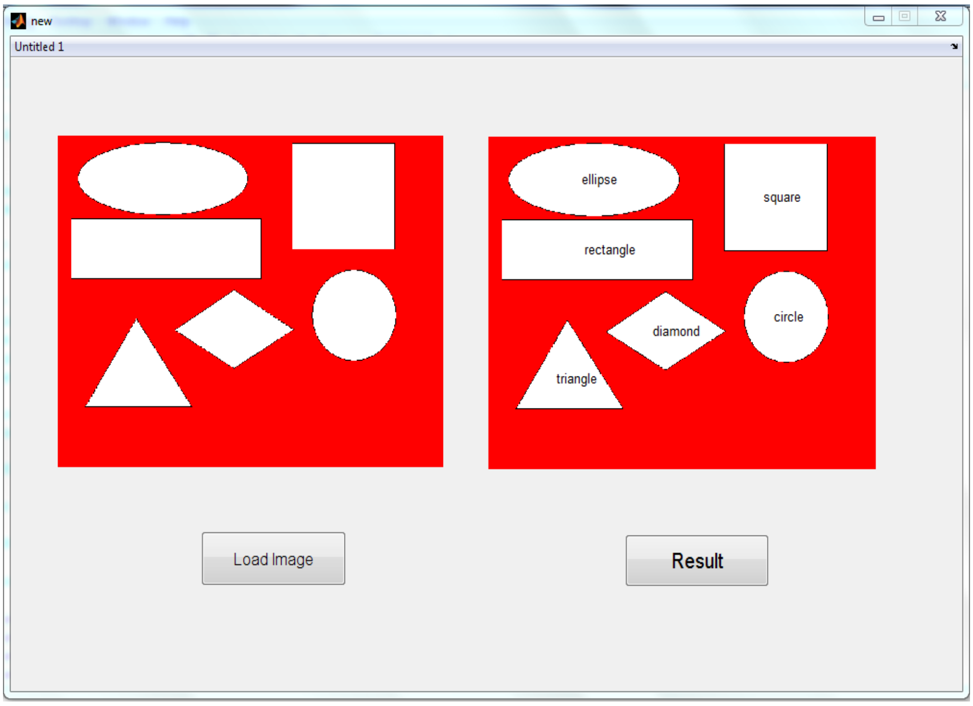

Fig.9 - Object Detection-Multiple objects

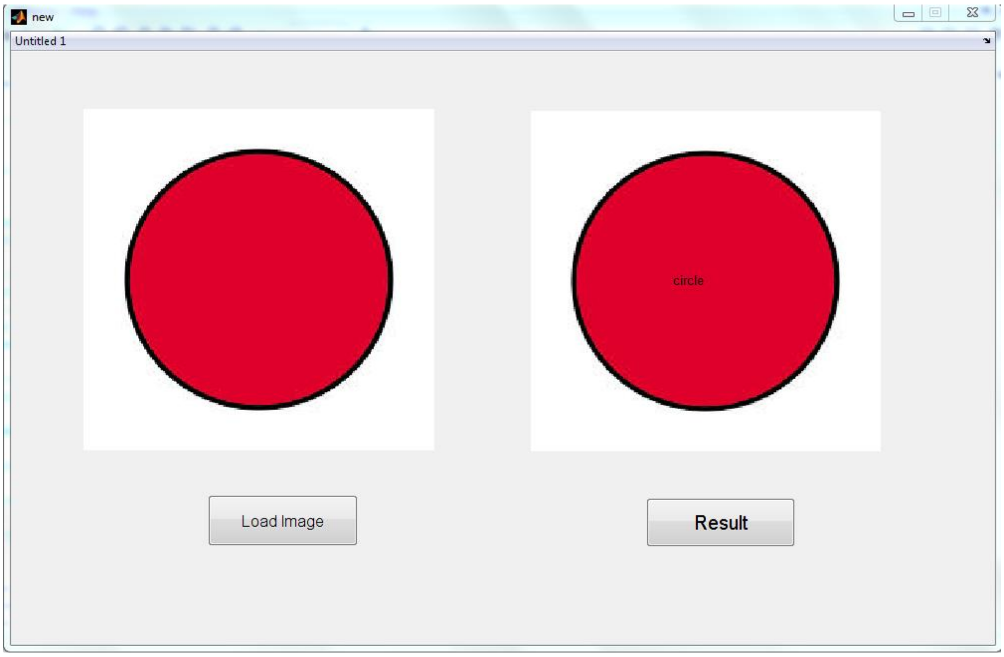

Fig.10 – Object detection-single object

6. CONCLUSION

There has been an extensive research work on paintbrushes in the past two decades. Paintbrushes were designed using printing techniques which were confined to smooth surfaces; few were designed to lessen the burden of repetitive work, while others to paint large surfaces mainly for interior wall painting in order to cut down on skilled labor. In our project we have come up with a unique concept of painting an object on any type of surface, detected in a captured image. Also where speed, accuracy and sketching intricate designs is of concern. The robot designed, can paint or sketch a design any number of times within no time. If we were to draw 50 symmetric circles manually we would take lot of time concentrating on dimensions and other factors, while this electronic paintbrush will be accurate as well as quick.

We have performed a thorough literature review and have found our project concept as a unique one. The MATLAB ® code for object detection was done using image processing techniques. MpLab software with C compiler was used to program PIC16F microcontroller and pickit2 to dump the program. A relay circuit board was used to build an efficient control system of robot.

7. RESULT

The electronic paint brush was successfully designed and implemented, which helps to achieve a painting device. We designed an electronic paintbrush with the help of PIC16F microcontroller and object detection algorithm in MATLAB®. The results of object detection are depicted in the above pictures. We interfaced a robot with microcontroller, which can draw simple images. The image of a circle or square is selected through MATLAB® and is recognized as a circle or square respectively and the signal is sent to the microcontroller through the serial communication port. The ARM is programmed to move in any particular direction with respect to the instruction received at serial communication port of microcontroller.

While the embodiments of the invention disclosed herein are presently considered to be preferred, various changes and modifications can be made without departing from the spirit and scope of the invention. The scope of the invention is indicated in the appended claims, and all changes that come within the meaning and range of equivalents are intended to be embraced therein.

ACKNOWLEDGEMENTS

This work was supported by the School of Electronics and Communication department at VIT University and by the Embedded systems department at VIT-Vellore Campus. The authors would like to express their sincere appreciation to these supports.

REFERENCES

[1] Anthonie H Bergman, Eindhoven (NL); Hubertus M.R. Cortenraad, Eindhoven (NL),"Electronic Paintbrush with Scanner and Dispensers". October 2010.

[2] Jonathan Louis Trent, Leonardtown, MD (US), "Hand Held Electronic Paintbrush". February 2004.

[3] P.Keerthanaa, K.Jeevitha, V.Navina, Automatic Wall Painting Robot, International Journal of Innovative Research in Science, Engineering and Technology Vol. 2, Issue 7, July 2013

[4] MATLAB and graphical user interfaces: Tools for experimental management. Behavior Research Methods, Instruments, & Computers 2000, 32 (2), 290-296

[5] Jayshree Sahu, S.K.Sahu, Jayendra Kumar, Microcontroller Based Dc Motor Control, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 3, May – 2012.