# Gesture Controlled Robot using Image Processing

Harish Kumar Kaura[1,] Vipul Honrao[2], Sayali Patil[3], Pravish Shetty[4],
Department of Computer Engineering
Fr. C. Rodrigues Institute of Technology, Vashi
Navi Mumbai, India

***Abstract*—Service robots directly interact with people, so finding a more natural and easy user interface is of fundamental importance. While earlier works have focused primarily on issues such as manipulation and navigation in the environment, few robotic systems are used with user friendly interfaces that possess the ability to control the robot by natural means. To facilitate a feasible solution to this requirement, we have implemented a system through which the user can give commands to a wireless robot using gestures. Through this method, the user can control or navigate the robot by using gestures of his/her palm, thereby interacting with the robotic system. The command signals are generated from these gestures using image processing. These signals are then passed to the robot to navigate it in the specified directions.**

***Keywords*— Gestures; OpenCV; Arduino; WiFly; L293D.**

## I. INTRODUCTION

In today's age, the robotic industry has been developing many new trends to increase the efficiency, accessibility and accuracy of the systems. Basic tasks could be jobs that are harmful to the human, repetitive jobs that are boring, stressful etc. Though robots can be a replacement to humans, they still need to be controlled by humans itself. Robots can be wired or wireless, both having a controller device. Both have pros and cons associated with them. Beyond controlling the robotic system through physical devices, recent method of gesture control has become very popular. The main purpose of using gestures is that it provides a more natural way of controlling and provides a rich and intuitive form of interaction with the robotic system. This mainly involves Image Processing and Machine Learning for the system or application development. Beyond this, it also requires some kind of hardware for interfacing with the system for gesture control. There are some systems that have been developed in the same field using various techniques.

### *A. Existing Systems*

Many systems exist that are used for controlling the robot through gestures. Some gesture recognition systems involve, adaptive colour segmentation [1], hand finding and labelling with blocking, morphological filtering, and then gesture actions are found by template matching and skeletonising. This does not provide dynamicity for the gesture inputs due to template matching. Another system uses machine interface device to provide real-time gestures to the robot [2]. Analog flex sensors are used on the hand glove to measure the finger bending [3], also hand position and orientation are measured by ultrasonics for gesture recognition [4]. And in another approach, gestures are recognized using Microsoft Xbox 360 Kinect(C) [5]. Kinect gathers the colour and depth information using an RGB and Infra-Red camera respectively. This system though is not very cost effective.

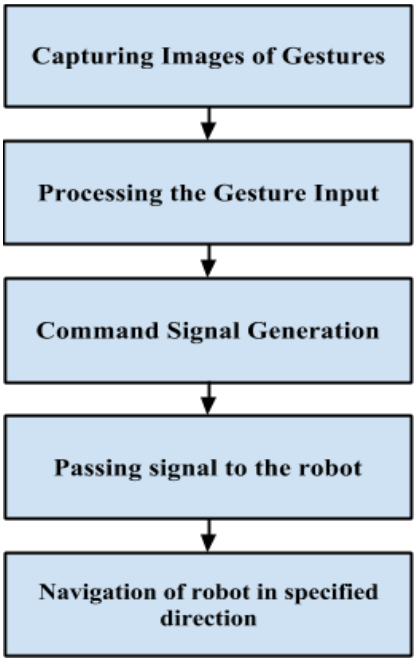


Fig. 1. Basic flow of the System

### *B. Proposed System*

We propose a system, using which the user can navigate the wireless robot in the environment using various gestures commands. The main objective is to provide reliable and a more natural technique for the user to navigate a wireless robot in the environment using gestures.

In this system, user operates the robot from a control station that can be a laptop or a PC with a good quality in-built webcam or external webcam. This webcam is used to capture real time video stream of hand gestures to generate commands for the robot. Gesture commands are given using hand palm. Mainly two kinds of gestures are used which are explained further. Irrespective of the gesture technique used, robot is moved in all possible directions in the environment using four possible types of commands which are Forward, Backward, Right and Left. Image frame is taken as an input and processed using Image Processing. Processed image is then used to extract the gesture command. This gesture command can have one of the four possible commands as specified. From this

generated gesture command, signal is generated to pass the given command to the robot.

Generated signal is stored in the file at the control station. Wi-Fi shield on the robot accesses this file to transmit the signals from the control station to the robot. As soon as the Wi-Fi shield gets command from the control station, it is passed to the Arduino microcontroller. Arduino takes this signal as input from the Wi-Fi shield and generates some output signals that are passed to the motor driver. This output signal generation depends on the gesture input, for every four possible gesture input, different output signal is generated. The motor driver is used to drive the DC motors of the robot. It takes digital signals as the input from the Arduino and gives these signals as an output to the DC motors. Once a command signal is given to the robot, it continues to move in that direction till the next command is given or any obstacle comes in the path. Figure 1 shows basic flow of the system.

## II. TECHNOLOGIES USED

### A. C++ with OpenCV:

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel. The library is platform independent. It focuses mainly on real time image processing and computer vision. OpenCV is written in C and its primary interface is C with wrapper classes for use in C++. [6] Also there are full interfaces available for Python, Java, MATLAB and Octave. It is used for recognition of gesture commands given by the user for the robot.

### B. Arduino: Duemilanove

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surrounding by controlling lights, motors and other actuators [7]. The microcontroller on the board is programmed using the Arduino Programming Language (based on Wiring) and the Arduino Integrated Development Environment (IDE) (based on processing). Arduino *B.* projects are stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

### C. Wi-Fi Shield: WiFly GSX:

The Wi-Fi Shield connects Arduino to a Wi-Fi connection. In the proposed system we used a Wi-Fi Shield called WiFly. It takes power from the Vin pin of Arduino which is regulated at 3.3V and is provided to main Wi-Fi chip called RN-131C.

Communication between Arduino and WiFly shield is over SPI using Arduino's digital pins 10-13 (CS, MOSI, MISO, and SCLK respectively). [8]

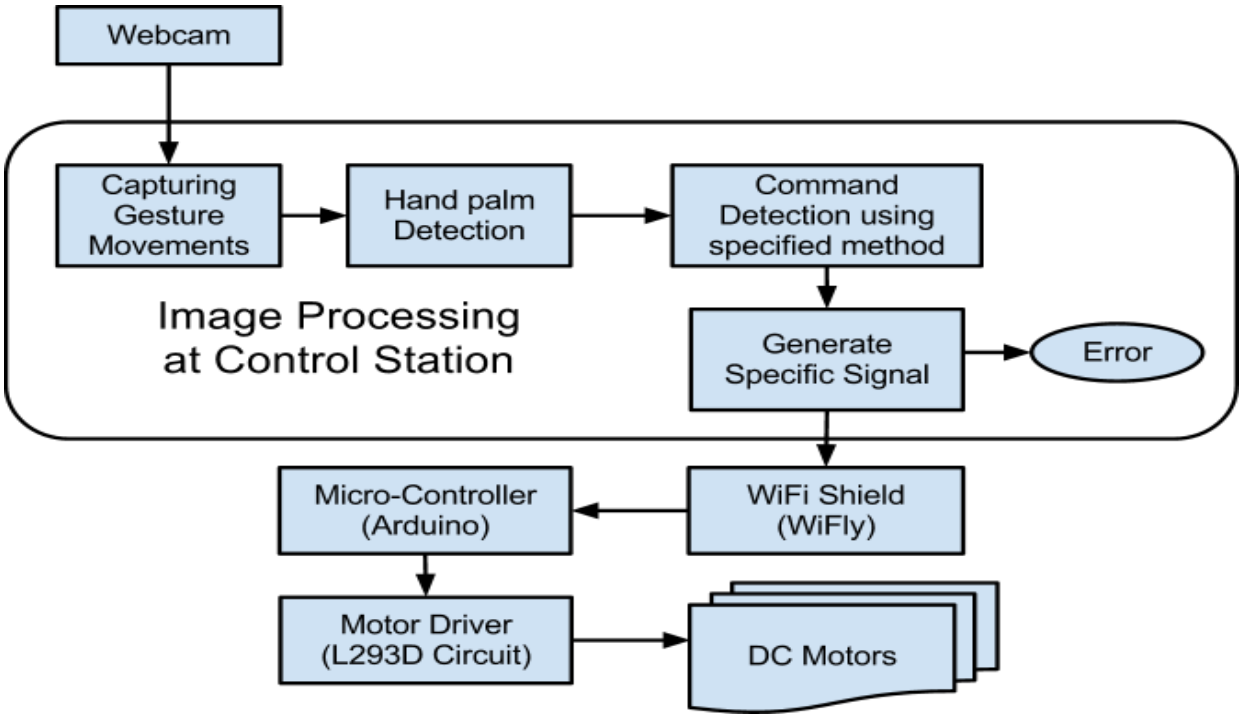


Fig. 2. Design of System


Fig. 3. Wi-Fi Shield (WiFly GSX)

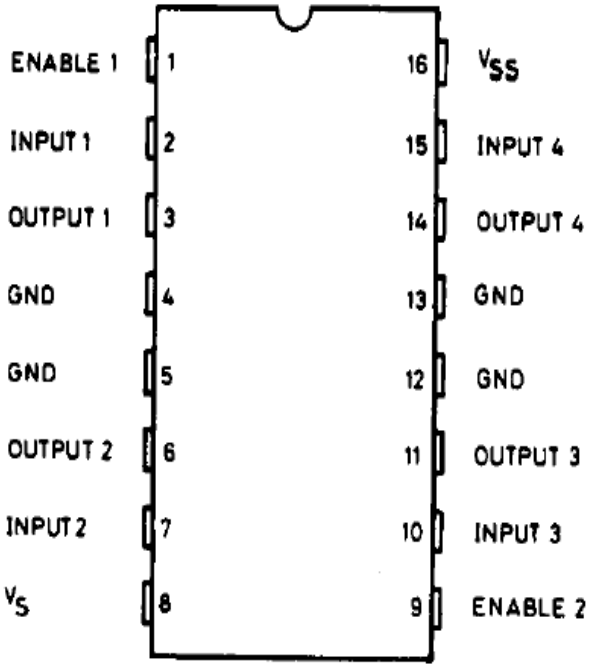


Fig. 5. Motor Driver (L293D)

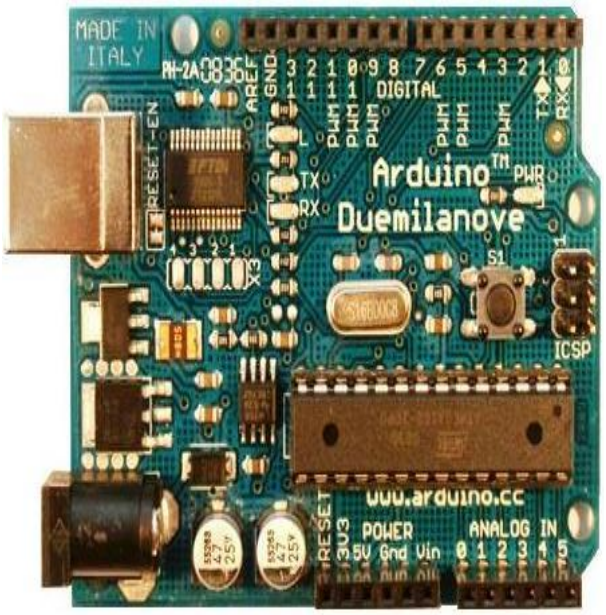


Fig. 4. Micro-Controller (Arduino-Duemilanove)

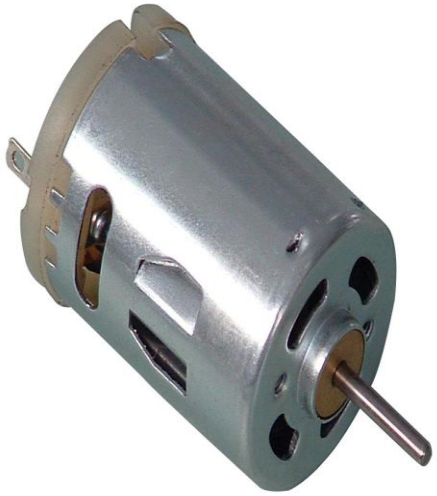

Fig. 6. DC Motor

### D. L293D: Motor Driver

The L293D is a monolithic, integrated, high voltage, high current and has 4 channel drivers. Basically this means using this chip one could use at most four DC motors and provide power supplies of up to 36V. The L293D chip uses H-Bridge. The H-Bridge is typically an electrical circuit that enables a voltage to be applied across a load in either direction to an output [9], e.g. motor. This means that just reversing the direction of current leads to reversing of the direction of motor.

### E. PHP:

PHP is an open-source server-side scripting language designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is printed by a Web server with PHP processor module which generates the resulting Web page.

## III. DESIGN

Our design primarily focuses on gesture command Recognition. Command is generated at the control station and sent to the robot through Wi-Fi within the Wi-Fi range. The robot moves in the specified direction according to the specified command. The following section illustrates the steps carried out in the design.

### A. Capturing Gesture Movements:

The end result of gesture recognition system is to generate a command and that is given to the robot. There are mainly four possible gesture commands that can be given to the robot (Forward, Backward, Right and Left). These gesture commands are given by the user in two ways. First method is based on the Finger Count and second is based on the direction given by the hand palm. Both methods involve recognition of the palm and process it further accordingly.

### B. Hand Palm Detection:

This involves two steps to detect Hand Palm, which are as follows.

#### 1) Thresholding of an Image Frame:

An image frame is taken as input through webcam. Binary Thresholding is then done on this image frame for the recognition of hand palm. Initially minimum threshold value is set to a certain constant. This value can be used to threshold an image and thus to increment the value till the system detects only one single blob of white area without any noise.

#### 2) Drawing Contour and Convex Hull:

After thresholding an image frame, contour is determined and drawn on the thresholded white blob. Number of contours generated will be many due to noise. So threshold value is incremented and same procedure is applied till one contour is generated. A convex hull is drawn around the generated contour points. Convexity defect is used to generate the list of defects in the convex hull. This is in the form of vectors that provides all the defect parameters which includes defect point in the contour, depth of the defect, start point and end point of the line in the convex hull for which defect has occurred.

### C. Command Detection using Specified Method

These 2 basic tasks are performed on the input frame. Depending on the kind of gestures further processing on an image frame is done. As stated earlier following are the two ways of controlling the robot through gestures:

#### 1) Finger Count based Gesture Control:

The convexity defect's list provides depth parameters for every defect. Defects with the highest depth are easily extracted from the list. These defects are basically gaps between two consecutive fingers. Finger count is determined as one more than the number of defects. For example, for one defect, finger count is two. Thus each count from two to five can be specified as a command for robot to move in the particular direction.

#### 2) Direction of Hand Palm Gesture Control:

In this method instead of giving a finger count, direction of the palm is used to determine the gesture command. Convexity defect specifies the depth point, start point and end point for the line of convex hull where the defect has occurred. Start and end points of the line specify fingertip. A depth point in the defect is the point in the palm where fingers meet. By comparing these point coordinate values, command is predicted. For this comparison the midpoint on the defect line is taken into consideration for calculations. This midpoint is then compared with the depth point. For a small difference in x coordinates and large difference in y coordinates, gesture command is predicted as forward and backward respectively. If the difference between midpoint and depth point is negative then gesture command is forward or else backward. The same thing is applied for a small difference in y coordinates and large difference in x coordinates. Instead of using all fingers of palm few fingers can also be used to specify the orientation.

### D. Generate Specific Signal:

A text file forms an interface between Control Station and the Robot. Gesture command generated at control station is written in the file with a tagged word. This is done using C++ fstream library. A particular value is written into the file for given command. To make robot move in forward, backward, right and left direction simple values are written as 1, 2, 3 and 4 respectively. For example, Forward command is represented as 1. Value written in a file for 1 is sig1 where sig is a tagged word in the file. As soon as there is a real time change in gesture command, file is updated. Similarly for the next command to turn right, value is written as sig3 in the file where sig3 is a tagged word in the file.

### E. Wi-Fi Shield: WiFly

WiFly works on HTTP port 80 for provided library. This library is used for communication of WiFly with the Control station through the Wi-Fi Hotspot. There is a simple web page present on the control station having PHP script. This is used for accessing the file and reading a command written in it.

WiFly on the robot connects itself with the Wi-Fi Hotspot. Then it continuously pings this web page on the control station. This is also called as Polling. From this, WiFly gets a command signal with a tagged word. This command signal is then transmitted to the Arduino without any modifications. Figure 3 shows WiFly GSX by Sparkfun.

*F. Micro-Controller: Arduino (Duemilanove)*

When Arduino gets command signal from the WiFly, it is having HTTP headers sent by web page with a tagged signal. Signal is read character by character and appended in the string. Every time after appending the character, Arduino checks for the tagged word in the string. For every iteration it checks the substring of tagged word at the end. If 'sig' is the tagged word in the signal, then programl check for substring sig at the end of a string in each iteration loop. As it gets a tagged word at the end of the string, it terminates the loop for reading the signal character by character. Then it reads only the next character which is an actual command signal generated by the gestures. Depending on this command, signal is sent to L293D motor driver through the digital pins of the Arduino. Four digital pins of the Arduino is set as input to the L293D PIC, two pins on both sides. It has four possible methods as forward(), backward(), right(), left(). Depending on the command signal, a particular method is called for every iteration. Each method is defined with a specified command to make each digital pin HIGH or LOW. Figure 4 shows Arduino of Duemilanove.

*G. Motor Driver: L293D Circuit*

It has four input pins two on each side of the PIC. All `these four pins are connected to the digital pins of an Arduino and four output pins are connected to DC motors of the Robot. Enable pins are used to enable input/output pins on both the sides of PIC and Vcc is used for supplying external power to the DC motors. Both the motors on the same side of the robot move in the same direction at a time. So positive ends of both motors are put in output pin 1 of PIC and negative ends of same motors are put into output pin 2, same thing is done for other side of PIC too. Vcc pin is be used to provide external power supply to the DC motors. Figure 5 shows L293D PIC.

*H. DC Motors:*

A DC motor is mechanically commutated electric motor powered from direct motor (DC). The stator is stationary in space by definition and therefore so is its current. The current in the rotor is switched by the commutator. DC motors better suited for equipment ranging from 12V DC systems in automobiles to conveyor motors, both which require fine speed control for a range of speeds above and below the rated speeds. The speed of a DC motor can be controlled by changing the field current [10]. Fig 6 shows most widely used DC motor.

## IV. IMPLEMENTATION

*A. Capturing Gesture Movements:*

Image frame is taken as input from the webcam on the control station and further processing is done on each input frame to detect hand palm. Figure 7 is an example of input frame. This involves some background constraints to identify the hand palm correctly with minimum noise in the image.

*B. Hand Palm Detection*

After capturing the image frame from the webcam, some basic operations are performed on this frame to prepare it for further processing of command detection. These operations are necessary for implementing both the techniques of gesture control, following two main processes are done to detect hand palm.

*1) Thresholded Image:*

Image frame taken as input from webcam is thresholded starting from minimum thresh value till single contour is formed in an image, same is in the case of intensity based thresholding. In the Figure 7 two fingers are shown by the user as a gesture command having dark background. That image is thresholded so that only a single contour can be formed on it. This thresholding is done on the basis of intensity in the image, which neglects the dark background and thresholds the fingers. Thresholded image is shown in Figure 8.

*2) Drawing Contour and Convex Hull:*

As shown in Figure 9, after obtaining thresholded image two main things are done, drawing the Contour on the thresholded part and fitting this contour in the Convex Hull. Contour is drawn in the thresholded image by using function drawContour() in the library OpenCV. This is done on the intermediate image, this image is then passed for drawing the convex hull. This covers the whole contour by joining the minimal points to form Convex Hull. These two basic operations are performed on every image frame taken from the webcam, and then depending on the kind of gesture technique chosen by the user, further processing on the images is done. These two techniques are Finger Count based gesture control and Direction of Hand Palm Gesture Control.

*C. Command Detection using Specific Method:*

After completion of pre-processing of an input frame, further processing is done on the extracted image according to specified technique. These two methods of giving gesture commands are as follows.

*1) Finger Count based Gesture Control:*

In this technique of giving gesture commands, first defects in the convex hull are found out using function convexityDefects(). Convex hull is formed using minimal set of points, so it does follow the contour path always, this causes the formation of defects in the convex hull. convexityDefects() function gives information about these defects stored in the form of vector.

This vector has values of each defect point in the coordinate format, end points of the line of defect, and depth of the defect point. The non-linear structure of the contour causes hundreds of defects for the convex hull. But the defect formed due to gap between two fingers has largest depth value as compared to other depth values of defects. Minimum threshold value is considered for the comparison with all depth values thus the depth values more than this threshold value would be the depth point of a defect, which would be formed due to gaps between two fingers. This is well applicable for the count two to five. As for the 1 such depth, count of finger will be 2 i.e. depth plus one and so on. In the Figure 10 two fingers are shown having only one depth point. So for that one depth point count is two. Also in Figure 11 same techniques are used to show a different command. After processing the image two defects are found out that specifies count as 3. This technique has some disadvantages so another technique can be used.

*2) Direction of Hand Palm:*

In the previous technique of Finger Count Gesture Control, image having no large depth fails. For example, for finger count one, there is no such large depth so it is difficult to recognise, as it is count one or there is no such count. So counts from two to five are used as command signals. In this technique of gesture command, orientation of hand palm is taken into the consideration for recognition of the command signal. Thus an orientation of hand palm gives direction in which robot is to be moved. This command can be given with minimum two fingers or with the whole palm.

In the Figure 12, gesture command of this technique is given using two fingers. Orientation of these two fingers is towards right side, so the command signal for right is generated and passed to the robot. For deciding orientation of the palm two things are used, depth point and end points of the line of defect. These parameters have been found out by initial processing of the gesture command recognition. Midpoint of line of defect is calculated by taking average of two end points. Then this midpoint's coordinate position is compared with depth point's position.

Each time two main conditions are checked for each frame. For small difference between y-coordinate and large difference between x-coordinate, possible commands are Right or Left. In Figure 12, depth point and midpoint has small y-coordinate difference and large positive difference between them, so orientation of fingers is towards right is predicted correctly.

For the difference between y-coordinate of the point, which is negative, orientation is in the left direction specifying command as left. Similarly in the above Figure 13, the orientation of two fingers is down, that is a command given to the robot is backward. In this image frame, there is a small change in x-coordinate and large positive change in y-coordinate of the depth point and midpoint of the line of defect. So the orientation of the two fingers is downwards, specifying command as backward. Similarly for the difference between y-coordinate of the point as negative, orientation is in the upward direction specifying command as forward.

*D. Generate Specific Signal:*

After detecting gesture command specific signal value is generated, unique for every gesture command. This signal value is written in the file using C++ file reading/writing functions.

*E. Wi-Fi Shield: WiFly*

As soon as the command is generated on the control station, it is written in a file with tagged word with it. This file is read by WiFly after regular interval. As it is a wireless communication, so WiFly communicates with the control station using a hotspot where control station is also in the same network. This hotspot can be a wireless router or any kind of Smartphone having tethering facility. Both control station and WiFly is provided with an IP address by using this IP address WiFly accesses the information from the control station using provided IP address of control station.

*F. Micro-Controller: Arduino- Duemilanove*

WiFly is connected to the Arduino through stackable pins shown in Figure 14. When the process of communication starts, WiFly tries to connect itself with the hotspot. For that it requires ssid and passphrase of the hotspot. These are provided in the burned code of the Arduino. After forming a successful connection with the hotspot, WiFly tries to get the access of the control station, with the provided IP address of the control station and port number of HTTP port which is by default 80.

As soon as WiFly gets connected to the control station, it continuously pings a PHP webpage on the control station which has a small PHP script, which returns the value of signal command written in the file with a tagged word. These received signal values are then passed to Arduino, which extracts command and calls which are specified for that command. Arduino sends four digital signals as an input to the L293D motor driver.

*G. Motor Driver: L293D*

L293D motor driver circuit is shown in the Figure 15. It takes digital signal as an input from the Arduino and gives digital output to the DC motors of the robot. Power supply to the circuit is given by rechargeable batteries. In this system some rechargeable mobile batteries are used as power supply each of 3.7V. To provide more voltage to drive the motors, 2-3 such batteries are connected in series.

*H. DC Motors in Robot:*

This is the final end product of robot consisting of all the hardware WiFly, Arduino and L293D motor Driver circuit on the robot chassis having power supply provided by the rechargeable batteries. Four DC motors are connected to this robot chassis as shown in Figure 16. This is controlled through gestures by the user at control station.

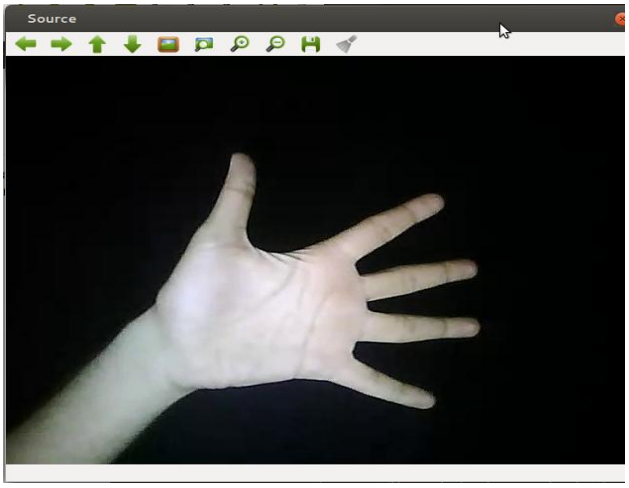


Fig. 7. Input Image Frame

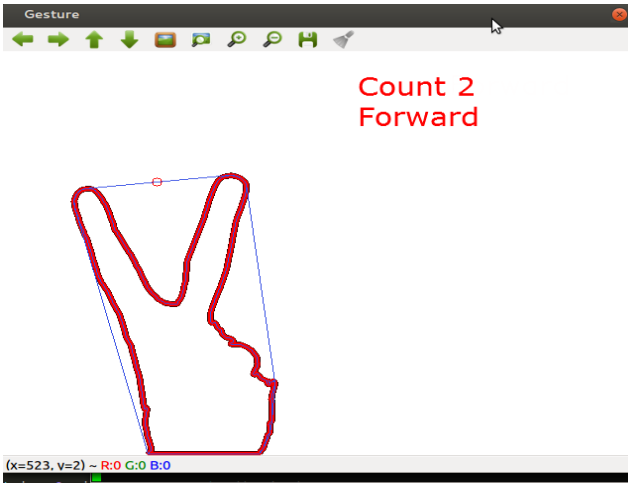


Fig. 10. Finger Count based Gesture Control I

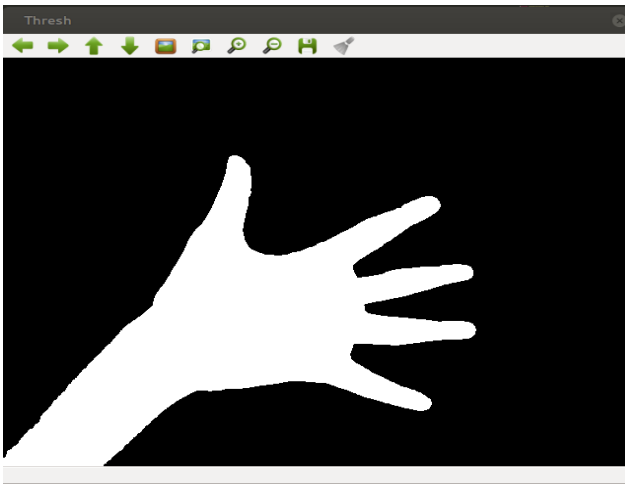


Fig. 8. Image Thresholding

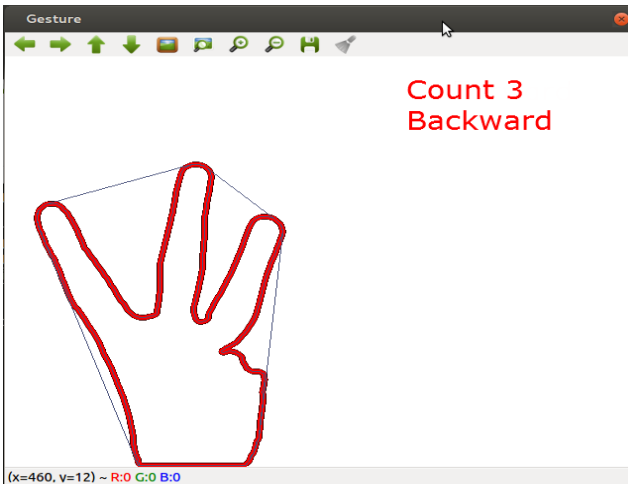


Fig. 11. Finger Count based Gesture Control II

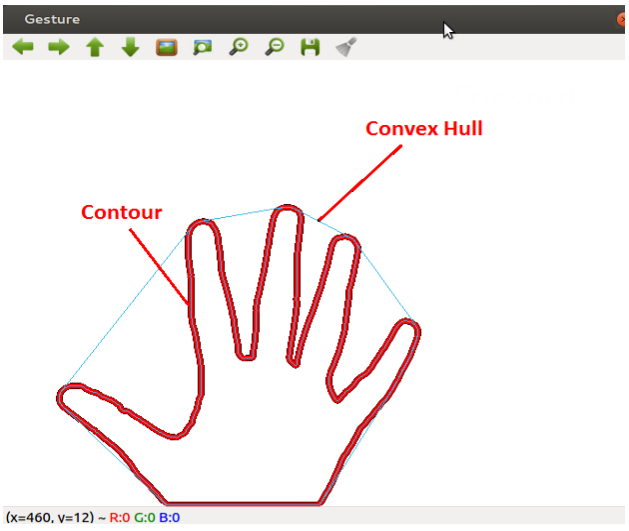


Fig. 9. Contour and Convex Hull

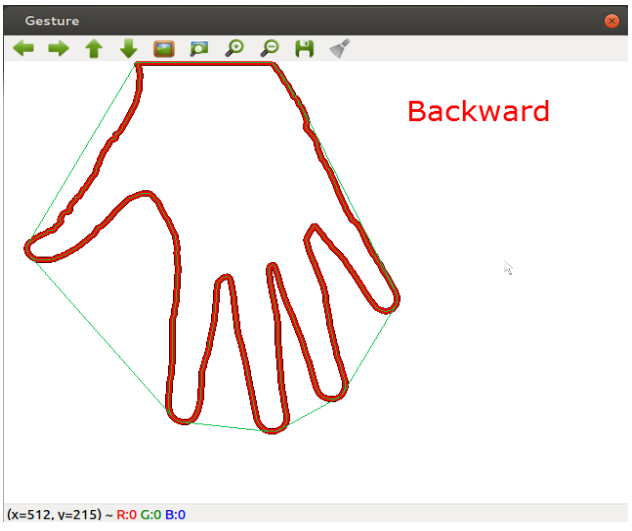


Fig. 12. Direction of Hand Palm I

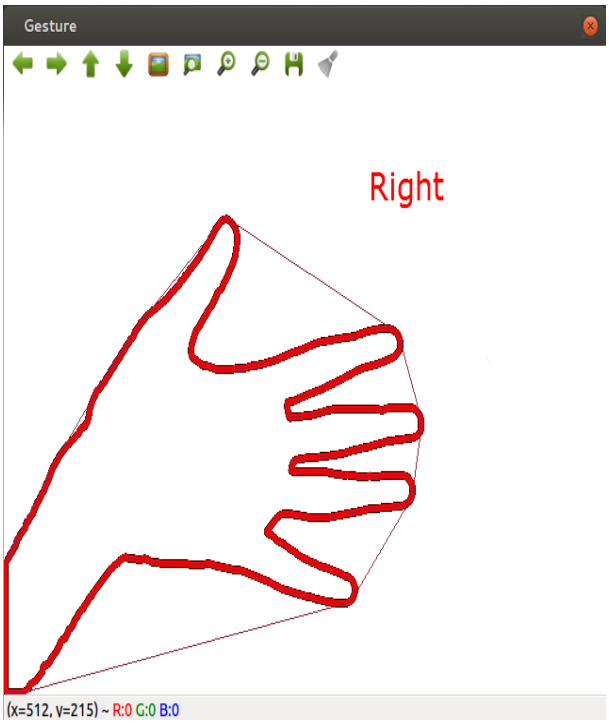


Fig. 13. Direction of Hand Palm II

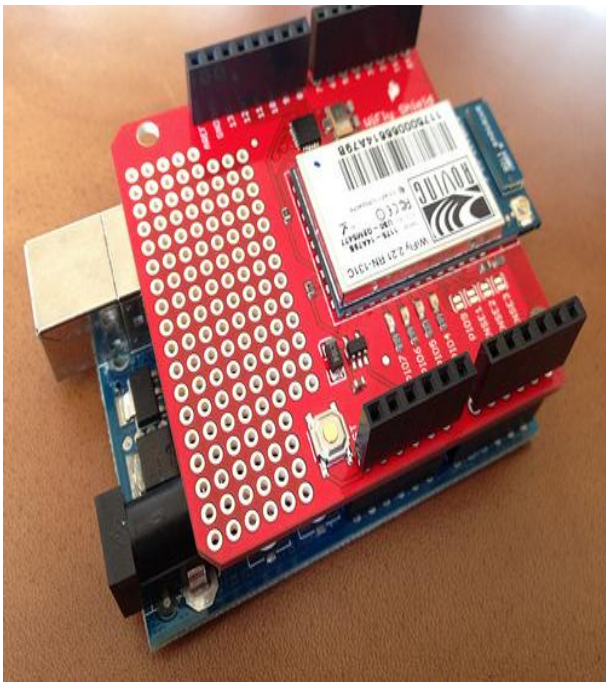

Fig. 14. WiFly interfaced with Arduino

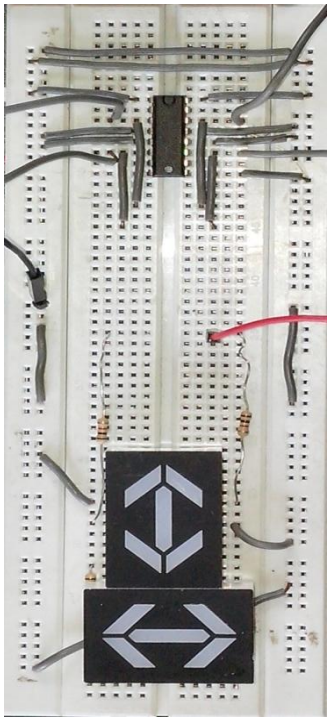

Fig. 15. Motor Driver- L293D Circuit

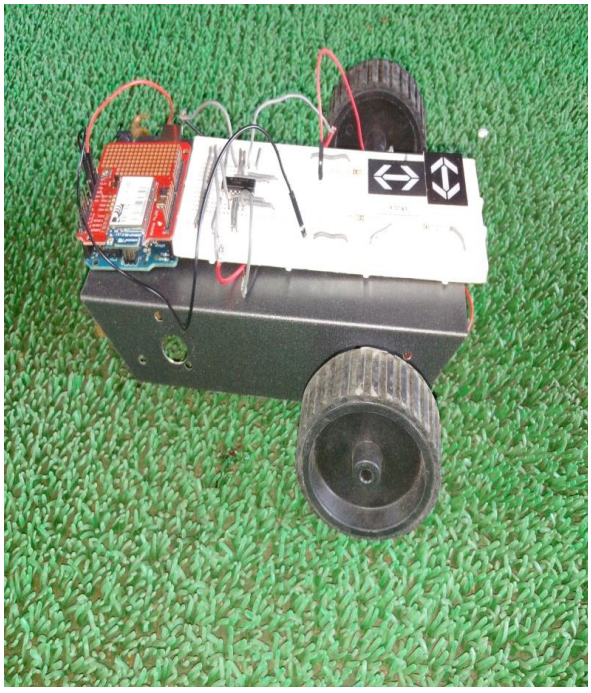

Fig. 16. Robot

## V. COMPARISION WITH EXISTING SYSTEM

The major advantage of our system over other systems is that it provides real time palm gesture recognition, leading to an effective and natural way for controlling robots. Additional advantages are:

- This implemented system is much more cost effective than the existing systems. As it does not involve any hardware requirement or configuration, there is little or no cost for the system's implementation. Moreover, ordinary webcams on PCs or laptops can be used for capturing gesture inputs.
- As mentioned earlier, it does not involve any specific hardware for gesture inputs - a normal webcam on laptop or PC can be used for gesture recognition. This system can be installed on any of these usable devices for gesture recognition. This provides flexibility to the user and the system is portable.
- The implemented system takes real-time gesture inputs from the user, processes these gesture inputs to generate command signals. For both methods of gesture input, processing is done by a method provided by the system, and it does not involve template matching to identify the finger count or direction of palm. Each image frame is processed and a command is generated in real-time. This provides higher accuracy for gesture recognition.

## VI. CONCLUSION

The Gesture Controlled Robot System gives an alternative way of controlling robots. Gesture control being a more natural way of controlling devices makes control of robots more efficient and easy. We have provided two techniques for giving gesture input, finger count based gesture control and direction of hand palm based gesture control.

In which each finger count specifies the command for the robot to navigate in specific direction in the environment and direction based technique directly gives the direction in which robot is to be moved.

At a time any one of the method can be used according to user's reliability, without using any external hardware support for gesture input unlike specified existing system. After gesture recognition command signal is generated and passed to the robot and it moves in specified direction.

## VII. FUTURE SCOPE

Presently in the system a minimum threshold value is set up and using that value input image frame is thresholded to binary. This approach put some constraints on the background, so a dark background is used. To avoid this, colour based thresholding can be done. According to user's hand palm colour, image can be thresholded within a tight bound limit. So hand palm can be easily detected irrespective of the background.

Also the current implementation makes use of periodic polling from WiFly to the web server to access the command signal in real time. This method of periodic polling may overload the server. Thus, instead of using periodic polling, a persistent connection between server and WiFly can be set up using HTML5's Web Socket API. Through this connection, the web server can push a command signal to WiFly asynchronously thereby reducing the load on the server.

## ACKNOWLEDGMENT

We would like to thank Ms. M. Kiruthika for supervising all activities and for lending her full cooperation, despite her busy schedule, in successful implementation of this system. We are also very grateful to Mr. Amiraj Dhawan and Mr. Rohit Jha for their guidance, help and support in the implementation.

## REFERENCES

[1] Chao Hy Xiang Wang, Mrinal K. Mandal, Max Meng, and Donglin Li, "Efficient Face and Gesture Recognition Techniques for Robot Control", CCECE, 1757-1762, 2003.

[2] Asanterabi Malima, Erol Ozgur, and Mujdat Cetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control", IEEE International Conference on Computer Vision, 2006.

[3] Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson and Young Harvill, "A Hand Gesture Interface Device", 189-192, 1987.

[4] Jagdish Lal Raheja, Radhey Shyam, Umesh Kumar and P Bhanu Prasad, "Real-Time Robotic Hand Control using Hand Gestures", Second International Conference on Machine Learning and Computing, 2010.

[5] Gesture Controlled Robot using Kinect http://www.e-yantra.org/home/projects-wiki/item/180-gesture-controlled-robot-using-firebirdv-and-kinect

[6] OpenCV Library http://docs.opencv.org/

[7] Arduino http://arduino.cc/en/Guide/HomePage

[8] WiFly Library https://github.com/sparkfun/WiFly-Shield

[9] L293D Motor Driver http://luckylarry.co.uk/arduino-projects/control-a-dc-motor-with-arduino-and-l293d-chip/

[10] DC Motors www.globalspec.com/learnmore/motion_controls/motors/dc_motors