

# Gesture Controlled Smart Bot using Image Processing

Christabel Rebello, Marshall Pereira, Roysten Rodrigues, Aarti Gokul

Computer Engineering, [christabel.rebello94@gmail.com](mailto:christabel.rebello94@gmail.com), 7768847725

**Abstract**— Robots are devices that mimic the appearance of a natural being. These robots interact with people directly. Thus making the interface user friendly. The early device was mainly based for navigation and controlling robot without any natural medium. To expedite a practical solution to this requirement we have implemented a system in which we will send commands to the robot using hand gesture method and we have also included an edge avoiding technique. With the help of this method the user can control the robot with his fingers. The image processing technique is used for capturing these commands. Thus making the robot move in required direction

**Keywords**— Arduino, OpenCV, Gestures, L293D Motor Driver Shield, IR sensors.

## INTRODUCTION

Robotic system is a widely increasing system in today's world. Robot is an artificial agent that helps in decreasing human efforts and increases efficiency and accuracy. Basic use of robots is to replace the harmful jobs and repetitive tasks that were to be done mainly by humans. In order to control the system with the help of any external hardware device in hand, the upcoming method of gestures is becoming popular which is operated without using any hand device. The gestures given by hand are making the robotic system user friendly and innate. Other than this, there is also some hardware device required for interfacing the gesture controlling and avoiding obstacle collision

## EXISTING SYSTEM

There are many existing system that are present in order to control robots with the help of gestures. Some of the gesture recognition system involves systems that mounts the camera in a cap worn by the user, system using depth perceptive sensors, hand position and orientation can be measured by ultrasonics for gesture recognition, in kinect for Xbox 360 the gestures recognized system is used.

## PROPOSED SYSTEM

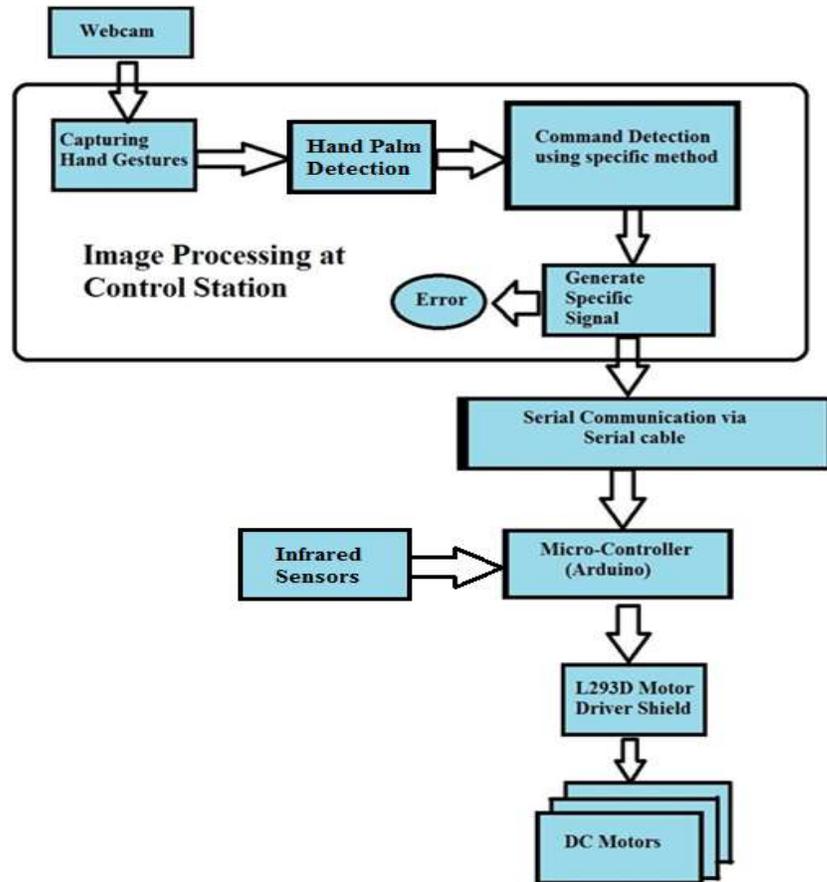
We proposed a system which uses a serial wire for navigation of a robot using hand gesture commands and we also used IR sensors for edge detection.[3] In this system the user can navigate a robot in four direction that is forward, backward, left and right as per required. In this robotic system, user can operate a robot from a controller station such as laptop with the help of inbuilt or an external webcam. Hand gesture commands are given using fingers. Firstly the image frame is been taken as input and then they are further processed. This processed input image is then used to extract command. After gesture command is obtained, the signal is generated to pass the command to the robot. The command signal that is obtained is stored in a file at the control station. Now the robot accesses the signals from the control station with the help of serial communication wire connected to the serial port of Arduino board and a control station. Arduino takes the signal as input from the control station with help of serial wire and generate some output that are passed to the motor driver. Different output signal is generated for four different input gestures. [1] Digital signals are taken by the motor driver as an input and output is given to the DC motors. In this system the robot detects the edge by sending infrared signal form the IR sensors.

## TECHNOLOGY USED

Arduino: Duemilanove.

The name Arduino describes a microcontroller development system. Arduino is open source component that may be individually hard to use. Arduino drops everything together into a mash-up of open source technologies when we try to give them the best user

experience to get something done quickly. The microcontroller on the board is programmed using the Arduino Programming Language and the Arduino Integrated Development Environment (IDE). You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment. Arduino are intended for making interactive projects. Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators. [11]



### 1. Design of System.

C++ with OpenCV :

OpenCV(Open Source Computer Vision Library) was designed for computational efficiency and with a strong focus on real-time applications. The code is written in optimized C++. OpenCV library is platform independent. The OpenCV library has more than 2500 optimized algorithms. Mainly these algorithms are used in face recognition, to capture movements, produce a high resolution image, find similar images in stored databases, also used in augmented reality concept, etc. Various interfaces that are included are C++, C, Python, Java and MATLAB. As it is platform independent it supports Mac OS, Linux, Android, Windows. [6]

L293D: Motor Driver:

The DC motors of our proposed robotic system can drive in either direction with the help of Motor Driver IC [5]. A single L293d IC can control two DC motors. Since they control two DC motors, L293D works on a concept of H-bridge. In order to make the voltage change its direction for rotating the motor in both clockwise and anticlockwise direction H-bridge IC are used by DC motor. Refer Fig. 2. [7]

IR sensors:

We find applications of infrared sensor in our day to day life. It helps in sensing certain attributes of environment by emitting infrared signals. It helps in avoiding obstacle as well as detecting edges to avoid collision. The wavelength of infrared wave ranges between 0.75 and 1000 $\mu\text{m}$ . Mainly the region between 0.75  $\mu\text{m}$  to 3  $\mu\text{m}$  is considered as near infrared region, where as the region between 3 and 6 $\mu\text{m}$  is known as the mid-infrared region and region beyond 6  $\mu\text{m}$  is known as far infrared region. Refer Fig. 2.



2. L293D Motor Shield.



3. Micro-Controller (Arduino-Duemilanove)



4. IR Sensor.



5. DC Motors.

## IMPLEMENTATION

The following section illustrates the steps carried out in the Implementation.

Taking input image frame from captured gestures: As soon as the gesture is made, the result is taken as input from the webcam on the control station and further processing is done on each input frame to detect hand palm and command is generated, which is to be given to the robot. Gestures are given by finger count method. There are four types of gesture commands created those are forward, backward, left and right.

Detecting Hand Palm: In order to detect hand palm we have used background subtraction model. There are different background subtraction models present in OpenCV, we are using the codebook method. It tries to subtract and calibrate for some time to be exact for some frames. This operation of detection is necessary for detecting gestures made by finger method. Hand palm detection involves following steps. [2]

1. Thresholding of an Image Frame:

The image frame is taken from the inbuilt or an external webcam of a PC or a laptop. Initially the image has a minimum threshold value. Gradually the threshold value is increased in order to obtain a clear image. The image frame is thresholded so that we can get a single contour free from noise. Thresholding neglects the dark background and thresholds the fingers. Refer to Fig. 6.

## 2. Drawing Contour and Convex Hull:

Now after acquiring the image frame that has been thresholded, contour is drawn by using functions from OpenCV library that is drawContour(). Following the contour drawing, now the convex hull is also drawn on that image. These two basic operations are performed on every image frame depending on the gesture technique given by the user. The gesture technique that we have used here is finger count method. [4]

Finger Count Gesture Control: The gestures are given by finger count method. Convexity defects provide depth parameters for every defects. The commands are generated from these defects. The defects are nothing but the gaps between two successive fingers. If there is one defect then the finger count will be two, if there are two defects finger count is three, if there are three defects finger count will be four, similarly if defects are four finger count will be five. These defects are found first in the convex hull using the function as convexityDefect(). The defects are stored in the form of vectors. This vector has a values of each defect point. Although there are many defects formed in the image frame, the defects that are formed due to the gap between successive fingers has the largest depth as compared to other depth values of defect found in the image. The main code for extracting the contour and detecting the convexity points is in the function given as

```
void detect(IplImage*img_8uc1,IplImage* img_8uc3). [9]
```

In this way the judgement has been made by these defects to make the robot move in possible four directions by sending commands to the robot. Refer Fig. 7, Fig. 8, Fig 9, Fig 10.

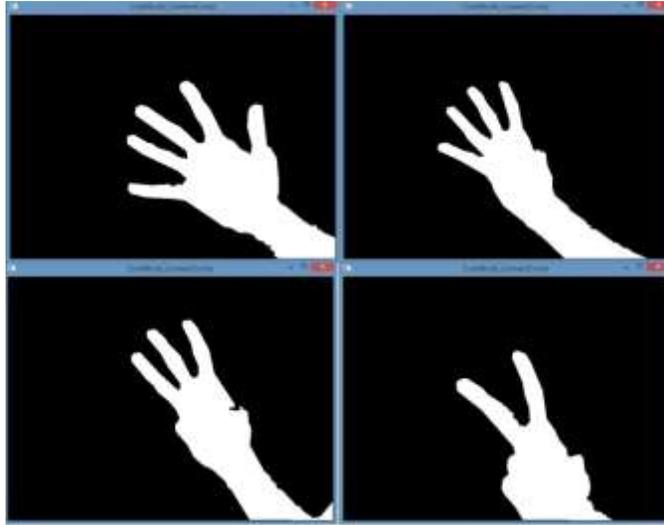
*Generating Signal from the command:* The command signal that is generated from the gesture is stored in a file on the control station. The signal is written in file using C++ functions. The value is written in a file for the respective command, to make the robot move. For example the forward command is represented as 1, backward command is represented as 2, left movement command is given as 3 and right movement command is given as 4. Since there is real time change in command, file is updated.

Passing of signal through serial wire to the Arduino: The generated command signal is read by the processing IDE at control station and then it is passed to the serial port of Arduino. Signal is read character by character and appended in the string. The Arduino reads the command signal generated by gestures. After this the command signal is send to the L293D motor driver through digital pins of Arduino. [12]

Motor driver: L293D: It is a circuit as shown in the figure FIG L293D IC takes an input from Arduino and sends digital output to the DC motors. We have given power supply to the motor driver with the help of two batteries of 9 volts each. Refer Fig. 2. [13]

DC motors: In our implementation DC motors are the end product. They take a digital input from motor driver L293D. Two DC motors are connected to the robot. [10]

Edge avoidance: IR sensors play a major role in implementation of the edge avoiding technique. Infrared waves from IR sensors are not visible to the human eye. In the electromagnetic spectrum, infrared radiation can be found between the visible and microwave regions. IR sensors work by using a specific light sensor to detect a light wavelength in the Infra-Red spectrum. We have mounted two IR sensors in the front part of the robot. The light from the LED bounces off the object and reflects towards the light sensors only when the object is close to the sensor. IR sensors continuously sense the ground, as soon as the sensor is unable to sense the ground, the robot comes to halt. Even if the gesture is given through the webcam, the robot won't move further, thus avoiding any collision to occur.



6. Binary Image Thresholding.



7. Finger Count based Gesture Control I.



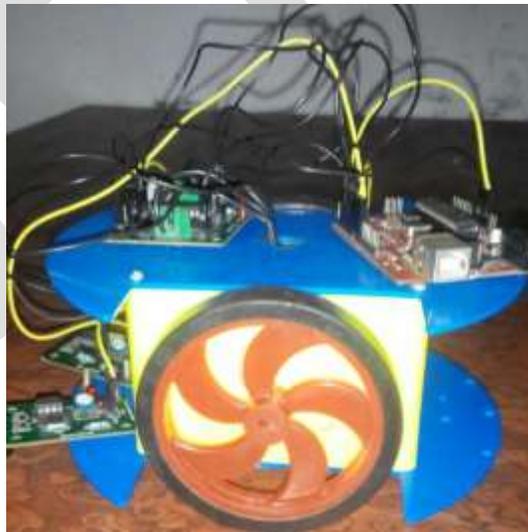
8. Finger Count based Gesture Control II



9. Finger Count based Gesture Control III



10. Finger Count based Gesture Control IV



11. Robot

#### ACKNOWLEDGEMENT

We would like to thank our project guide Mrs. Aarti Gokul who has been a source of inspiration and her insight and vision has made it possible for us to go on and work for this project. Her patience, encouragement, critique and availability made this dissertation possible.

We are grateful to the authorities, faculty and staff of Xavier Institute of Engineering who have helped us to better acquainted with the recent trends in technology and from whom we have learned so much. A special thanks to Mr. Stalin Dharamraj and Mr. Omprakash Yadav for their technical support and guidance. Moreover, we are thankful to our friends and families who have been patient and supportive to undertake this project.

#### CONCLUSION

An innovative and alternate system has been achieved through gesture controlled robot mechanism. More intuitive and natural as well as easy mode of control is achieved through gesture controlling system in order to control robots in an efficient and exciting way. We have used a specific method of giving inputs to gestures that is finger count method. Here commands are given to the robot which is derived from finger counts so as to make the robot move in desired directions. From the two methods any one method can be chosen and used at a time depending on the user. This eliminates the use of external hardware support (For eg. Remote Controllers) for gesture input unlike the existing system.

#### 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. Teemu Savolainen, Jonne Soinen, and Bilhanan Silverajan
- [6] OpenCV Library <http://docs.opencv.org/>
- [7] L293D Motor Driver <http://luckylarry.co.uk/arduino-projects/control-adc-motor-with-arduino-and-l293d-chip/>
- [8] DC Motors [www.globalspec.com/learnmore/motion\\_controls/motors/dc\\_motors](http://www.globalspec.com/learnmore/motion_controls/motors/dc_motors)
- [9] [anikettatipamula.blogspot.ro/2012/02/hand-gesture-using-opencv.html](http://anikettatipamula.blogspot.ro/2012/02/hand-gesture-using-opencv.html)
- [10] Vivek Hanumante, Sahadev Roy, Santanu Maity "Low Cost Obstacle Avoidance Robot", ISSN: 2231-2307, Volume-3, Issue-4, September 2013
- [11] Arduino <http://arduino.cc/en/Guide/HomePage>
- [12] [www.ladyada.net/learn/arduino/lesson4.html](http://www.ladyada.net/learn/arduino/lesson4.html)
- [13] Arduino Controlled L293D Robot, <http://www.instructables.com/id/Arduino-and-L293D-Robot-Part-1-/>