# Gesture Controlled Robot

Shaunak Chokshi, Shashank Sharma, Hardik Joshi

**Abstract**— Even after more than decades of input devices development, many people still find the interaction with the computers and robots an uncomfortable experience. Efforts should be made to adapt computers and robots to our natural means of communication: speech and body language. The aim of our project is to implement a real time command system through hand gesture recognition, using general purpose hardware and low costing sensor like a simple raspberry-pie and an USB webcam, so any user can make use of it in industries or at home. The basis of our approach is a fast segmentation process to obtain the hand gesture from the whole image which is able to deal with large number of hand shapes against different backgrounds and lightning conditions, and a recognition process that identifies the hand posture for different control application

**Index Terms**— Raspberry Pie, Convex hull, PWM, Motor diver, Contour, Beagle bone, Contoller board

## 1 Introduction

Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus there's a need of building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse. This could potentially make conventional input device such as mouse, keyboards and even touch-screens redundant. The project is an application for live motion gesture recognition using Raspi camera module and performs the action corresponding to it. In our case we have controlled the motion of a mobile robot according to the gesture of the user.

## 2 Block Diagram

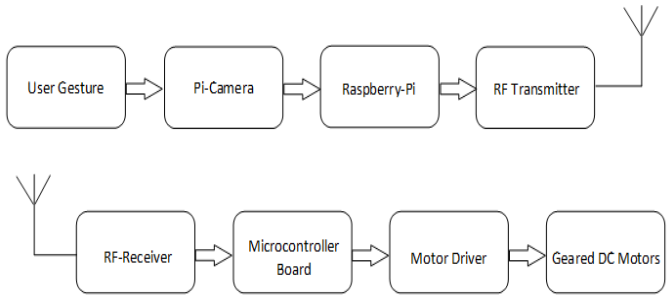


Fig. 1. General Block Diagram.

## 3 Hadware

Implementing image processing using hardware is a cumbersome and difficult task thus selection of the proper development board becomes an important issue. The most commonly used boards with which we all are familiar are the Ardiuno, Raspberry Pi and Beagle Bone.

| Name | Arduino Uno | Raspberry Pi | BeagleBone |
|---|---|---|---|
| Model Tested | R3 | Model B | Rev A5 |
| Price | $29.95 | $35 | $89 |
| Size | 2.95"x2.10" | 3.37"x2.125" | 3.4"x2.1" |
| Processor | ATMega 328 | ARM11 | ARM Cortex-A8 |
| Clock Speed | 16MHz | 700MHz | 700MHz |
| RAM | 2KB | 256MB | 256MB |
| Flash | 32KB | (SD Card) | 4GB(microSD) |
| EEPROM | 1KB | | |
| Input Voltage | 7-12v | 5v | 5v |
| Min Power | 42mA (.3W) | 700mA (3.5W) | 170mA (.85W) |
| Digital GPIO | 14 | 8 | 66 |
| Analog Input | 6 10-bit | N/A | 7 12-bit |
| PWM | 6 | | 8 |
| TWI/I2C | 2 | 1 | 2 |
| SPI | 1 | 1 | 1 |
| UART | 1 | 1 | 5 |
| Dev IDE | Arduino Tool | IDLE, Scratch, Squeak/Linux | Python, Scratch, Squeak, Cloud9/Linux |
| Ethernet | N/A | 10/100 | 10/100 |
| USB Master | N/A | 2 USB 2.0 | 1 USB 2.0 |
| Video Out | N/A | HDMI, Composite | N/A |
| Audio Output | N/A | HDMI, Analog | Analog |

Fig. 2. Comparission between different development boards.

### 3.1 Development Board

We choose to work on raspberry pi due to its high clock speed and cost-effectiveness. Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S700 MHz processor, Video Core IV GPU, and has 512 MB of RAM. It has a Level 1 cache of 16 KB and a Level 2 cache of 128 KB. The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with C, C++, Java, Perl and Ruby.

### 3.2 Communication

We have used RF-module, WIR-1186 available on Robokits India Pvt. Ltd. for transmitting data from laptop to robot as RF has long

International Journal of Scientific & Engineering Research Volume 7, Issue 6, June-2016
ISSN 2229-5518

communication range. This module integrates RF69, an extremely low-power sub-GHz transceiver, an MCU for wireless network control, data handling and hardware interface, a PCB antenna and matching circuit. It supports UART communication protocol and baud rate ranging from 9600bps to 115200bps.

### 3.3 Driving Of Robot

We have used two single channel motor driver (model no. RKI-1340) available at Robokits India Pvt. Ltd. for controlling the speed of the motors used in robot. This driver is 6v-24v compatible 20A capable DC motor driver. It comes with a simple TTL/CMOS based interface that can be connected directly to I/Os of a MCU. Speed of the motor can be controlled by PWM signals generated by any MCU.

## 4 Gesture Recognition Algorithm

The aim of hand detection is to device a program that is able to detect the hands, track them in real time and perform some gesture recognition. It was done with simple signal processing performed on images obtained from a Raspberry Pi camera. It detected the human hand from the image and detect it as a right hand or left hand. It included the following steps to obtain the results.

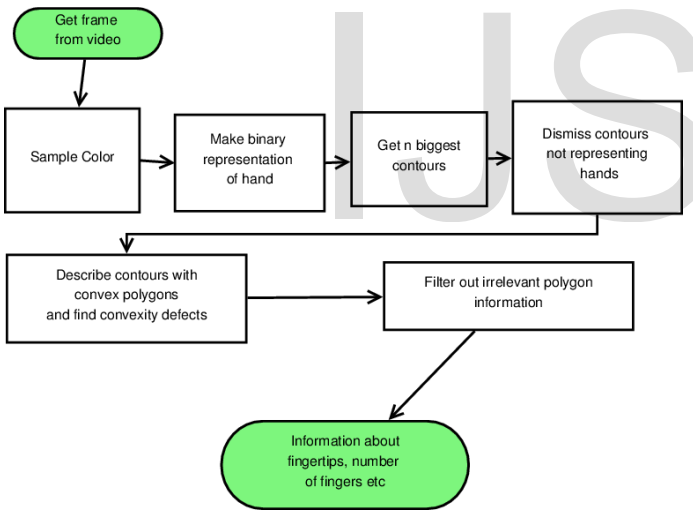


Fig. 3.Block Diagram of Hand Detection.

### 4.1 Detecting Background

From the feed from the camera, we remove the background. We use running average over a sequence of images to get the average image which will be the background too.

$$C[i][j] = \propto CurBG[i][j] + (1-\propto)CurFrame[i][j]$$

Fig. 3.Equation to detect background by averaging.

This equation works because of the assumption that the background is mostly static. Hence for those stationary item, those pixels aren't affected by this weighted averaging and $\propto.x + (1-\propto) = x$. Hence those pixels that are constantly changing isn't a part of the background, hence those pixels will get weighed down. Hence the stationary pixels or the background gets more and more prominent with every iteration while those moving gets weighed out. Thus after a few iterations, we get the above average which contains only the background. In this case, even face is a person is a part of the background as it needs to detect only my hands.

### 4.2 Background Subtraction

A simple method to start background subtraction is we can subtract the pixel values. However this will result in negative values and values greater than 255, which is the maximum value used to store an integer, but if we have a black background then nothing gets subtracted in that case. Instead we use an inbuilt background subtractor based on a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. Background subtraction involves calculating a reference image, subtracting each new frame from this image and thresholding the result which results in a binary segmentation of the image which highlights regions of non-stationary objects. We then use erosion and dilation to enhance the changes to make it more prominent.

Fig. 5.Output of background subtraction

International Journal of Scientific & Engineering Research Volume 7, Issue 6, June-2016
ISSN 2229-5518

### 4.2 Contour Extraction

Contour extraction is performed using OpenCV's inbuilt edge extraction function. It uses a canny filter.

Fig. 6.Output of contour detection

### 4.3 Convex Hull and Defects

By having the set of points for the contour, we find the smallest area convex hull that covers the contours. We observe that the convex hull points are most likely to be on the fingers as they are the extremities and hence this fact can be used to detect no. of fingers. But since our entire arm is there, there will be other points of convexity too. So we find the convex defects i.e, between each arm of the hull, we try to find the deepest point of deviation on the contour.

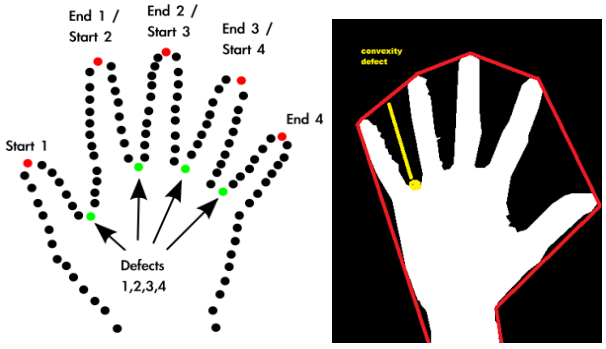


Fig. 7.Output of Convex Hull and Defects detection

### 4.4 Tracking and Finger Detection

The defect points are most likely to be the center of the finger valleys as pointed out by the picture. So we find the average of all these defects which is definitely bound to be in the center of the palm but it's a very rough estimate. So we average out and find this rough palm center. As assumption is to be made that the palm is angled in such a way that it is roughly a circle. So to find the palm center , we take 3 points that closes to the rough palm center and find the circle center and radius of the circle passing though these 3 points. Thus we get the center of the palm. Due to noise, this center keeps jumping, so to stabilize it, we take an average over a few iterations. Thus the radius of the palm is an indication of the depth of the palm. Using this we can track the position of the palm in real time and even know the depth of the palm using the radius.

The next challenge is to detect the no. of fingers. We use a couple of observations to do this. For each maxima defect point which will be the finger tip, there will be 2 minimal defect points to indicate the valleys. Hence the maxima and the 2 minimal defects should form a triangle with the distance between the maxima and the minimas to be more or less same. Also the minima should be on or pretty close to the circumference of the palm. We used this factor too. Also the ratio of the palm radius to the length of the finger triangle should be more or less same. Hence using these properties, we get the list of maximal defect points that satisfy the above conditions and thus we find the no of fingers using this. If no of fingers is 0, it means the user is showing a fist.

## 5 Output

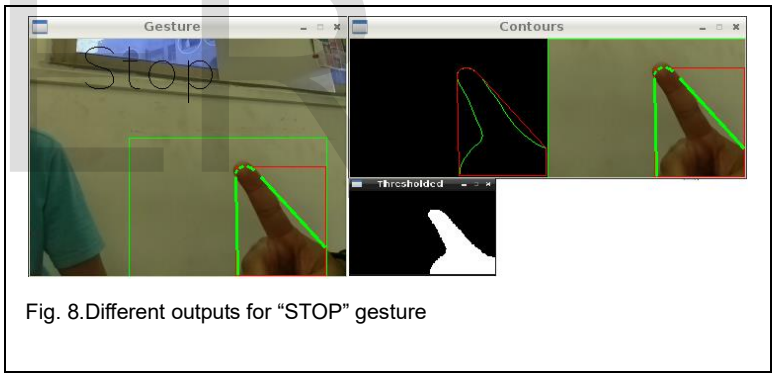


Fig. 8.Different outputs for "STOP" gesture

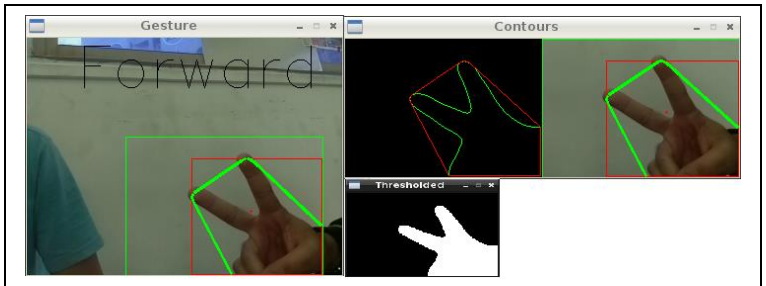


Fig. 9.Different outputs for "FORWARD" gesture

International Journal of Scientific & Engineering Research Volume 7, Issue 6, June-2016
ISSN 2229-5518

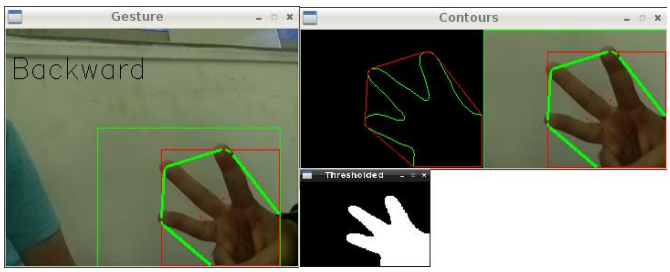


Fig. 10.Different outputs for "Backward" gesture

## 6 Conclusion

We have implemented Convex-Hull detection method to recognize the gesture in raspberry-pi. There are many algorithms available to detect gesture like Haar Cascade method where we need a lot of positive and negative images to train the filter which is very time consuming and adding a new gesture would be thus very cumbersome. However, In the Convex-Hull detection algorithm we can easily add a gesture and since this algorithm uses adaptive thresholding, there is no effect of light intensity and it sets the threshold dynamically with the amount of light present. The drawback of this method is the limited number of gesture as gestures depend on the number of fingers.

## References

[1] http://www.intorobotics.com/9-opencv-tutorials-hand-gesture-detection-recognition/
[2] http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0
[3] http://stackoverflow.com/questions/6471023/how-to-calculate-convex-hull-area-using-opencv-functions
[4] http://www.swarthmore.edu/NatSci/mzucker1/opencv-2.4.10-
[5] http://www.robopapa.com/projects/InstallOpenCVOnRaspberryPi
[6] http://eyalarubas.com/face-detection-and recognition.html
[7] Daniel Lelis Baggio, Shervin Emami, David Millan Escriva, Khvedchenia Levgen, Naureen Mahmoon, Jason Saragih and Roy Shilkrot, "Mastering OpenCV with Practical Computer Vision Projects", Packt Publishing Ltd., First Edition- December 2012.
[8] Gesture recognition: Enabling natural interactions with electronics by Texas Instruments
[9] Uma Sahu, Ditty Varghese, Gayatri Gole, Melanie Fernandes, and Pratiksha Mishra, "Hand Cursor Implementation Using Image Processing & Sixth Sense Technology", International Journal of Infinite Innovations in Technology- ISSN: 2278-9057 IJIIT, Volume-I Issue-III 2012-2013, January Paper-08 Reg. No.:20121208 DOI: V1I3P08.
[10] Amiraj Dhawan, Vipul Honrao, "Implementation of Hand Detection based Techniques for Human Computer Interaction", International Journal of Computer Applications (0975 – 8887),Volume 72, Page No.17, June 2013.