Obstacle Avoidance and Handgesture Recognition using Raspberry Pi and OpenCV

Mannem Sri Nishma, A. Sri. Nagesh

[1]M.Tech Scholar, Department of CSE, RVR & JC College of Engineering, AP, India.

[2]Professor, Department of CSE, RVR & JC College of Engineering, AP, India.
[1]srinishma@gmail.com

Abstract

Human-Computer Interaction (HCI) with gesture recognition is meant to acknowledge a variety of meaningful human expressions and has become a valuable and intuitive computer input technique. Hand gestures are one of the foremost intuitive and customary sorts of communication. Vision-based hand gesture recognition has received a big amount of research attention in recent years. However, the sector still presents variety of challenges for researchers. The vision-based hand gesture interaction process between humans and computers, gesture interpretation must be performed quickly and with high accuracy. A low-cost HCI system with hand gesture recognition is proposed. Skin and motion detection is employed for capturing the region-of-interest from the background regions. A connected component labeling algorithm is proposed to spot the centroid of an object, to spot the precise area of hand gesture, the arm area is removed with the help of a convex hull algorithm. The main aim of this system is to spot the gesture from imaging devices and convert them into corresponding text. The hand gesture algorithm that embeds in Raspberry Pi is employed to navigate a mobile robot implemented to urge a vision-based robotic system that depends on human-machine interaction. The Mobile robot moves to avoid the whole obstacles in its path using the Ultrasonic sensor.

Keywords: *Computer Vision, Raspberry Pi, Gesture recognition, Python, Ultrasonic Sensor*

1. Introduction

The Internet-of-Things (IoT) enables to regulate and communicate with computer systems. The IoT age has seen the growing importance of efficient human-computer interaction (HCI). The applications of HCI are versatile, starting from signing, through medical rehabilitation to a computer game. Especially for a few virtual environment applications, HCI offers the chance to integrate various new technologies to supply a more immersive experience for users. HCI and robot vision are active research areas in recent years. For various HCI systems, hand gesture-based HCI could be the foremost natural, convenient and intuitive method of communication between people and machines since it closely mimics how humans interact with one another. Its intuitiveness has produced many applications for exploring large and sophisticated data in computer games.

Gesture recognition has been a really interesting problem in the Computer Vision community for an extended time. This is often, particularly because the segmentation of the foreground object from a cluttered background may be a challenging problem in real-time. The foremost obvious reason is due to the semantic gap involved when a person looks at a picture and a computer watching a similar image. Humans can easily find out what's in a picture except for a computer, images are just 3-dimensional matrices. Gestures may represent a single command or a sequence of commands, a single word, or a phrase and can be static or dynamic. The Recognition system should be accurate enough to provide the correct classification of hand gestures in a reasonable time. Human-robot interaction using hand gestures provides a formidable challenge. This is because the environment contains a complex background, dynamic lighting conditions, a deformable hand shape, and a real-time execution requirement.

We are going to recognize hand gestures from a video sequence. To recognize these gestures from a live video sequence, we first need to take out the hand region alone removing all the unwanted portions in the video sequence. After segmenting the hand region, we then count the fingers shown in the video sequence to instruct a robot based on the finger count. Thus, the entire problem could be solved using 2 simple steps –firstly, find and segment the hand region from the video sequence. Later the second step is to count the number of fingers from the segmented hand region in the video sequence. Based on the finger count the mobile robot is navigated. Graphical User Interface is designed to access and control the mobile robot using various methods of communication like SSH, Bluetooth and using Camera module. The block diagram of the hand gesture recognition system is shown in fig1.

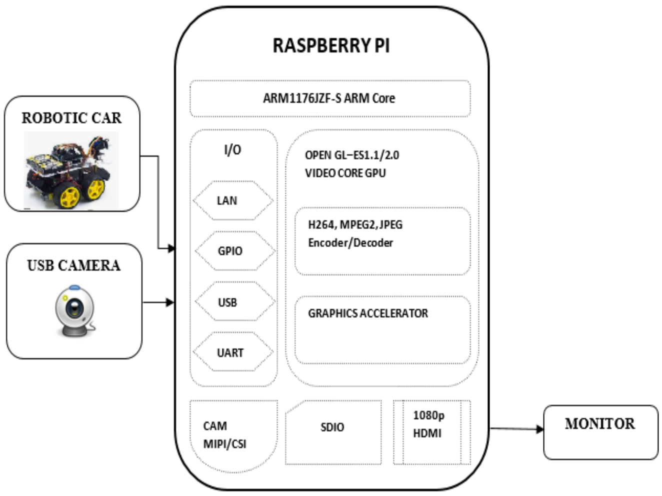

Fig 1: Block diagram of Gesture Recognition System

Raspberry pi is a credit card-sized computer board developed by the Raspberry pi foundation. It gets plugged in a TV or monitor and provides a fully functional computer capability. It computes data and produces valuable outputs, and controls components in a system based on the outcome of its computation.

2. Related Work

Francis et al [1] compared methods for gesture recognition in cars, evaluating glove-based, accelerometers-based and Kinect-based approaches. Mitra et al[2] analyzed more computationally high methods using hidden Markov models and finite state machines. Ghotkar et al [3] presented a unique approach for segmentation of hand and recognizing gestures using different color spaces.

The methods proposed by Francis et al required additional hardware, while those proposed by Mitra et al were computationally heavy, require more classification and processing time. Our goal was hence to follow the instance of Ghotkar et al and explore the more basic methods of hand segmentation and gesture recognition approaches available and applying them to execute simple controls on a laptop. Ribeiro and Gonzaga [4] proposed different methodologies for real-time GMM (Gaussian Mixture Method) background subtraction algorithms using video sequences for image segmentation.

Stergiopoulou and Papamarkos [5] proposed the YCbCr segmentation. A real-time hand posture recognition using 3D range data analysis is proposed by Popa, D., Simion, G., Gui, [6]. Mariappan [7] uses a motion detection algorithm for the recognition of gestures. Popa et al. [8], proposed trajectory-based hand gesture recognition using kernel density estimation and the related mean shift algorithm. Bugeau and Perez [9] proposed a method for detecting and segmenting foreground moving objects in complex scenes using clusters.

3. Proposed Methodology

Most of the entire hand interactive systems are often considered to be composed of three layers: detection, tracking, and recognition. The detection layer is liable for defining and extracting visual features which will be attributed to the presence of hands within the field of view of the camera(s). The tracking layer is liable for establishing temporal data association between successive image frames so that, at each moment in time, the system could also be conscious of what's where. Moreover, in model-based methods, tracking also provides how to take care of estimates of model parameters, variables, and features that are not directly observable at a particular moment in time. Last, the recognition layer is liable for grouping the spatiotemporal data extracted within the previous layers and assigning the resulting groups with labels that are related to particular classes of gestures.

There are many applications like media player, MS-Office, Windows picture manager, etc. which require a natural and intuitive interface. Nowadays most of the users use a keyboard, mouse, pen, Joysticks, etc. to interact with computers, which aren't enough for them. Within the near future, these existing technologies which are available for computing, communication and display are going to become a bottleneck and therefore the advancement in these technologies will be required to form the system as natural as possible.

The main objective of this work was the development of a control system for mobile robot movement, based on gesture recognition. A chip and robust control system without using data gloves or colored gloves, or other devices this system is developed. We used a generic webcam for the image acquisition process. The gesture recognition algorithm makes it suitable for real-time control, easy to implement, and efficient in unconstrained environments. This paper describes the hand gesture recognition algorithm for HCI and its evaluation on a specific application.

This system proposed and designed for a vision-based hand gesture recognition system contains various stages and the working flow-chart of the system is shown in fig2.

3.1. Image Acquisition

The first module in the proposed system is the image acquisition module. A live video is captured using a webcam. The recorded videos are of resolution 640x480 pixels. The recorded video is then converted into frames. These frames are named serially as per their sequence.

3.2. Pre-processing

Image processing techniques are used to improve the standards of the image frames. First, the hand region is extracted using a bounding box. The RGB image is converted into a grayscale image and further is converted into a binary image.

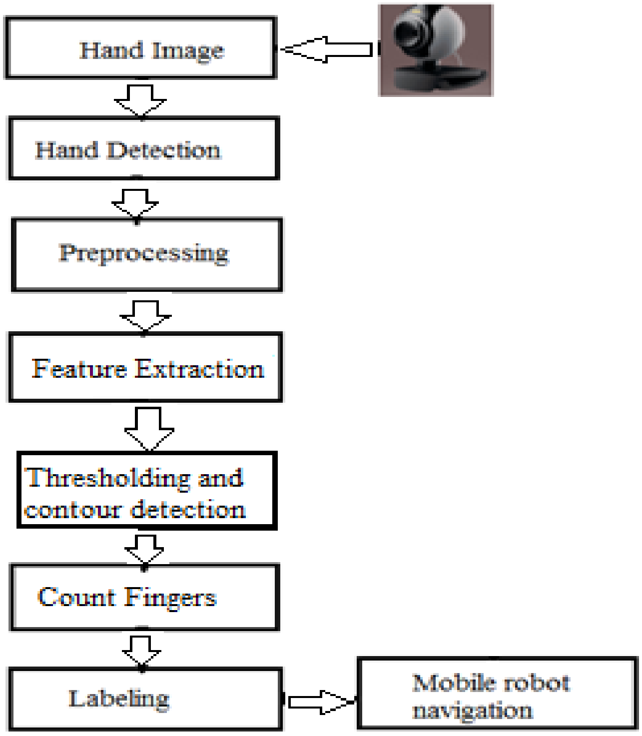

Fig 2: Flow chart of Hand Gesture Recognition System

3.3. Thresholding and Contour Detection

Thresholding is applied using Otsu's method to detect the region of interest (ROI). The Gaussian filter is employed for blurring the image. This step is applied for the removal of high-frequency noise and smoothing. After segmenting the hand region, we then count the fingers shown within the video sequence to instruct a robot based on the finger count. To detect the hand region from this difference image, we'd like to threshold the difference image, so that only our hand region becomes visible and all the other unwanted regions are painted as black. Thresholding is the assignment of pixel intensities to 0's and 1's based on a specific threshold level so that our object of interest alone is captured from the captured image.

After thresholding, we find contours in the resulting image. The contour with the largest area is assumed to be our hand. Contour is the outline or boundary of an object located in an image. Find the convex hull of the segmented hand region (which is a contour) and compute the most extreme points in the convex hull (Extreme Top, Extreme Bottom, Extreme Left, Extreme Right). The convex hull is a dynamic, stretchable envelope that wraps around the object of interest (hand). Fig 5 describes various phases of the hand gesture recognition system. For obstacle avoidance, an Ultrasonic sensor is used. In case of any obstacle detected in the path, the direction of the mobile robot gets changed. The mobile robot chooses the alternative path.

4. Implementation and Applications

A. Import necessary packages like NumPy, cv2 and raspberry pi import RPi.GPIO as GPIO package.
B. Define the GPIO pins of the raspberry pi to connect it with a mobile robot.
C. Initialize the current frame of the video.
D. Capturing is done as follows:
For frame in cam.capture_continous (rawCapture, format = ”bgr”.use_video_port = TRUE)
E. Convert frame to grey-scale.
F. To remove noise from the image by applying Blur, Gaussian Blur (GBlur) and Threshold the image. Thresholding is the assignment of pixel intensities to 0’s or 1’s based on the threshold level. This is done to identify the object of interest.
G. Find Contours using cv2.findContours () method.
H. Find Convex hull using cv2.convexHull().
I. Draw contours using cv2.drawContours().
J. Count the fingers and display the labels.
K. Find distance using ultrasonic sensor as follows:

```
Dist =   round (dist,2)
avgdistance =   avgdistance + dist
if  avgdistance  <   15:
        stop()
        backward()
else:
        forward()
```

4.1. Blur and GBlur

Blurring is employed for smoothing frames and reducing noise and details from the frame. With blurring, the smooth transformation from one color to another and the reduction of the edge contents are satisfied. Mathematically, applying a Gaussian blur to a picture is that the same as convolving the image with a Gaussian function. This is also referred to as a two-dimensional Weierstrass transform. A Gaussian blur has the effect of reducing the image's high frequency components. Gaussian blur is a low pass filter. Gaussian blur is a sort of image blurring filter that uses Gaussian function. The equation of the Gaussian function is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

There are various applications of the gesture recognition system, some of them are Sign Language recognition, Virtual Controllers, Remote Control, to aid physically Challenged people and Gaming.

5. Hardware and Software Description

5.1. Hardware Requirements

The hardware components of the system consist of Raspberry Pi 3 Model B+ as shown in fig 3, Camera Board, 5 inches 800*480 Resistive HD Touch Screen, L298 H-bridge driver, Rover four-wheel-drive platform, rechargeable batteries as a power supply and Ultrasonic sensor.

Fig 3: Raspberry Pi 3 Model B+

5.2. Software Platform:

Implementation of this gesture recognition system is carried out using Open CV and Python on the Intel core TM i5 Processor. Raspbian OS is needed.
Programs needed for the system are summarized as follows:

5.2.1. Raspbian OS (Operating System)

Raspbian OS has been developed for Raspberry Pi. This system has over 35000 packages, which are all set to backup the Raspberry pi environment. It is free and can be downloaded from the internet (known as NOOPS) and then copy it into a 16GB (or more) RAM stick.

5.2.2. Python and OpenCV

Python is a well-known and high-level programming language used for general-purpose programming. It is created in 1991. It allows programmers to express concepts in fewer lines of code than possible in languages such as C or Java. Python is a dynamic system and supports automatic memory management system and multi-programming style is supported including object-oriented, functional programming, and also procedural programming. Besides that, it has large and comprehensive standard libraries, Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.

OpenCV is a free library that includes hundreds of APIs for computer vision used in image processing to optimize a real-time application. There are many features in OpenCV that support data processing, including camera calibration, object detection, 3D reconstruction and interface to video processing. The primary interface of OpenCV is written in C++ but it supports other interfaces also such as C, Python, Java and MATLAB. In this paper python programming language is used with the required libraries from OpenCV to build the hand gesture recognition system that controls the motion of a mobile robot by avoiding obstacles during the movement using an ultrasonic sensor.

6. Experimental Results

To control the four DC motors of the mobile robot, first, it connects each motor to A (Out 1 & Out 2) and B (Out 3 & Out 4) connections on the L298N module. Next, it is powered by two 9V batteries. The Raspberry Pi is powered from the smart supply with 5V.

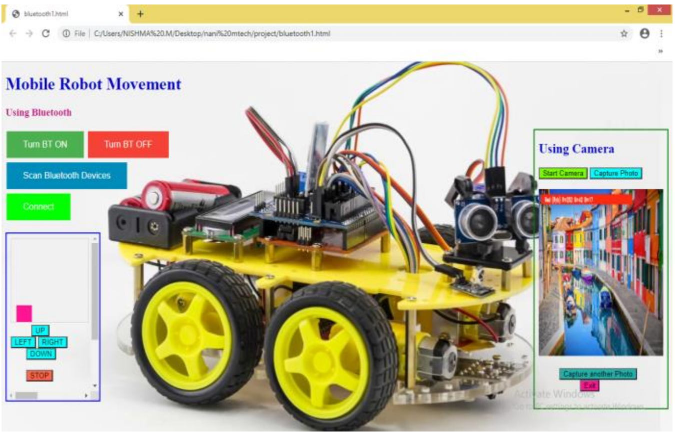

Fig 4: GUI for mobile robot navigation

Six GPIO pins are needed on Raspberry Pi, GPIO10 to enable motor A, GPIO09 to enable motor B, and the input pins (IN1, IN2, IN3 and IN4) of the L298N driver are connected with (GPIO22, GPIO18, GPIO16 and GPIO12) of Raspberry Pi respectively. The Ultrasonic sensor consists of four pins. Vcc is connected to GPIO 5V (pin 2), GND is connected to GPIO GND (pin 6), TRIG is the output pin and ECHO is the input pin. The direction of motors is controlled by the hand gesture recognition system. The designed system is navigated in four directions: Forward, Backward, Left, Right movements in addition to Stop. The recognition rate of the designed system is reached 98%. The overall system costs about 200$ and proved a good operation when navigated in a clear environment.

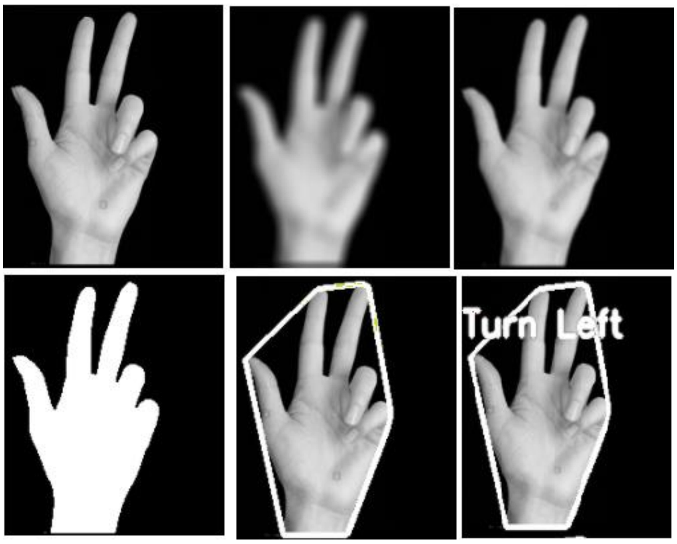

Fig 5: (a) Captured image (b) Blur image (c) GBlur image
(d) Threshold image (e) Contour Detected (f) Labeled image.

7. Conclusion and Future Work

This paper presents a quick, robust and accurate method for hand gesture recognition under unconstrained scenes. Experimental results show satisfactory recognition percentage of the gestures considered. The failure of the system to recognize the gesture is mainly because of the very changeable lighting conditions and moving objects (persons) entering the scene, the operator's failure to move the hand to the proper posture. It must be emphasized that after brief experience operators get used to the system. Future work will be focused on algorithm improvement, by employing a combination of segmentation techniques and robot motion control by tracking the arm movement and its speed.

Acknowledgments

The authors are greatly acknowledged to DST-FIST (Government of India) for funding to setting up the research computing facilities at RVR&JC College of Engineering.

References

[1] *J. Francis and A. B K, "Significance of Hand Gesture Recognition Systems in Vehicular Automation-A Survey", International Journal of Computer Applications, vol. 99, no. 7, pp. 50-55,* ***(2014).***

[2] *S. Mitra and T. Acharya, "Gesture Recognition: A Survey", IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 3, pp. 311- 324,* ***(2007)****.*

[3] *A. S.Ghotkar and G. K. Kharate, " Hand Segmentation Techniques to Hand Gesture Recognition for Natural Human-Computer Interaction", International Journal of Human-Computer Interaction, vol. 3, no. 1, pp. 15-25,* ***(2012)****.*

[4] *Hebert Luchetti Ribeiro, Adilson Gonzaga, "Hand Image Segmentation in Video Sequence by GMM: a comparative analysis", Symposium on Computer Graphics and Image Processing (SIBGRAPI'06),* ***(2006).***

[5] *E. Stergiopoulou and N. Papamarkos, "A new technique for hand gesture recognition", Proceedings of IEEE International Conference on Image Processing, Atlanta, pp. 2657-2660, Oct. 8-11,* ***(2006)****.*

[6] *Malassiotis, S. and Strintzis, M. G., "Real-time hand posture recognition using range data," Image Vision Comput. 26(7), 1027-1037* ***(2008)****.*

[7] *R. Mariappan, "Video Gesture Recognition System," ICCIMA, vol. 3, pp.519-521, International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007),* ***(2007)****.*

[8] *Popa, D., Simion, G., Gui, V., and Otesteanu, M. 2008. "Real-time trajectory-based hand gesture recognition". WSEAS Trans. Info. Sci. and App. 5, 4* ***(Apr. 2008)****, 532-546.*

[9] *Bugeau, A. and Pérez, P. 2009. "Detection and segmentation of moving objects in complex scenes". Comput. Vis. Image Underst. 113, 4* ***(Apr. 2009)****, 459-476.*

[10] *Raspberry Pi 3 Model B datasheet, RS Company, www.rs-components.com/raspberrypi.*

[11] *Ahmad, Tohari, Hudan Studiawan, and T. Ramadhan. 2014. "Developing a Raspberry Pi-based Monitoring System for Detecting and Securing an Object". International Electronics Symposium (IES), pp. 125-129,* ***(2014)****.*

[12] *Hemalatha, P., CK Hemantha Lakshmi, and S. A. K. Jilani, August 2015. " Real-time Image Processing based Robotic Arm Control Standalone System using Raspberry Pi". SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), Volume 2, Issue 8, pp. 18-21,* ***(2015)****.*

[13] *Thomas, Ron Oommen, and K. Rajasekaran, April 2014. " Remote control of robotic arm using raspberry pi". International Journal of Emerging Technology in Computer Science &Electronics (IJETCSE), Volume 8, Issue1, pp.186-189,* ***(2014)****.*

[14] *Tepljakov, Aleksei, 2015. "Raspberry Pi based System for Visual Object Detection and Tracking". Bachelor's Thesis,* ***(2015)****.*

[15] *P. Garg, N. Aggarwal and S. Sofat, "Vision based hand gesture recognition", World Academy of Science Engineering and Technology, vol. 49, no. 1, pp. 972-977,* ***(2009)****.*