www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 3 March 2015, Page No. 10952-10954

An Implementation of Hand Gesture Recognition System for Controlling a Car Buggy

[1]Sanjyot Agureddy, [2]Mohammed Esoofally, [3]Meenakshi Raut, [4]Manish Bhutada

Savitribai Phule Pune University,
Dept. Of Computer Engineering,
Trinity College of Engineering & Research, Pune, India
sanjyot.agureddy@hotmail.com[1]
mzeesoofally@gmail.com[2]
meenaraut0703@gmail.com[3]
manish.n.bhutada@gmail.com[4]

Abstract: ***In today's world, the use of hand gestures to control devices has increased significantly. We see the use of hand gestures in application like controlling a presentation, mobiles, gaming, toy-cars, direction systems, automotive engines etc. A new trend has been emerging for many human based electronic products, because of the gesture recognition system. Therefore the need to study the gesture recognition system becomes a must. In our paper we are proposing a model, where we are trying to control a car buggy using hand gestures. The hand gestures will be recognized using a Near Infrared Camera (NIR). This can also be implemented using an ordinary low resolution camera. The better the camera, better the results. This system is known as the Human Vehicle Interaction System (HVI). The HVI possesses the following characteristics, namely like robustness, Stability & Reliability. In our paper we propose the use of series of algorithms to process the image and perform desired action.***

Keywords: *Hand Gestures, Gestures Recognition System, Human Vehicle Interaction System, Image Processing, RGB Model, HSV model, Thresholding, Center of Gravity.*

1. Introduction

In the last few decades, a significant role has been played by the gesture recognition system. Technology has undergone major changes, technology has transformed from using remote controls to using hand gestures to control devices. In simple words we can put hand gestures as making hand movements. This technology has also aided the life of disabled in day to day life. People who suffer from lower limb injury can use hand gestures to set the wheelchair in motion to move from one place to another without anyone's help.

The Gesture recognition System has 3 stages into it. They are namely as follows:-

1. Capturing the Object
2. Processing the Image
3. Perform Action

In the first stage, the hand gestures are tracked and they are captured using a camera in low resolution of 320 x 240. In the second stage, the captured image is processed with a series of algorithms, to reduce the noise and recognise the gesture. In the final stage, based on the processed image, the desired action is performed according to the recognized gesture.

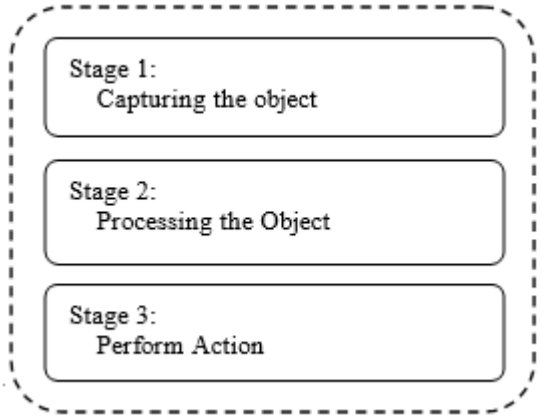

Figure 1: Gesture recognition Stages

The captured images cannot be directly used as medium to control the system. The captured images has to go through a few transformations. The captured image is in a 24bit format, on applying various transformations the image is converted to 1 bit format.

The series of transformation it is subjected to is as follows:-

1. Image Grab
2. Image Blur
3. RGB to HSV conversion
4. HSV Thresholding
5. Blob Detection
6. Gesture Recognition
7. Perform Action

2. Literature Survey

The current Gesture recognition system consist of Glove-based methods, bare hands, or controller based methods.
The Glove based method make use of delicate gloves connected with wires and sensors. Which can be damaged very easily if subjected to pressure. It is also risky as the wire can come in contact of the person and making him prone to shock. The controller based method, make use of a controller or a Kinect kit (from Xbox). Thus making the system dependent on the controller and it is also an expensive equipment. Finally the bare hand system, does make the use of hand which is inexpensive but while recognizing a gestures an external entity behind the actual user making any hand movement causes the system to possess incorrect results.

3. Proposed System

In this paper we are trying to control a car buggy using hand gestures. The gestures are captured using a camera. The captured image has to go through various transformations (various image processing algorithms). The transformed images gives the gesture. Based on the position of the hand on the screen the gesture is identified. The location of the color pointer on the screen helps it to predict the action that is predefined within the system. The color of the band is used to set the threshold value, which will be used for tracking the gestures. There is no specific restriction on the color that is to be used for tracking. The colors are to be acknowledged to the system before using the system. Color band are used in our system instead of bare hand because, bare hand tend to match up with physical color surrounding of the area, which results in wrong guessing and also degrades the working of the system. The system flow is shown in the below figure.

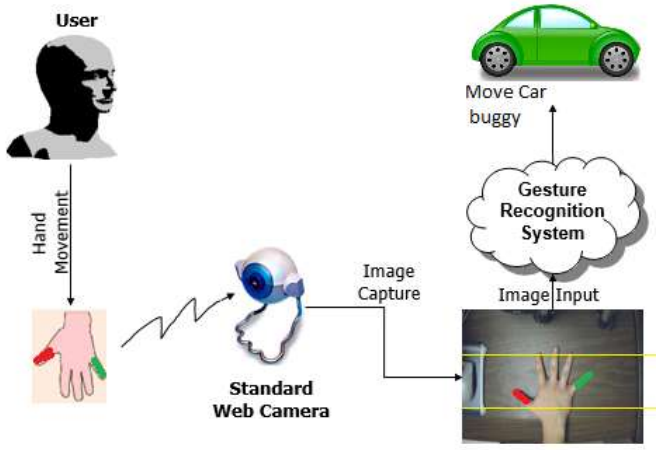

Figure 2: System Overview

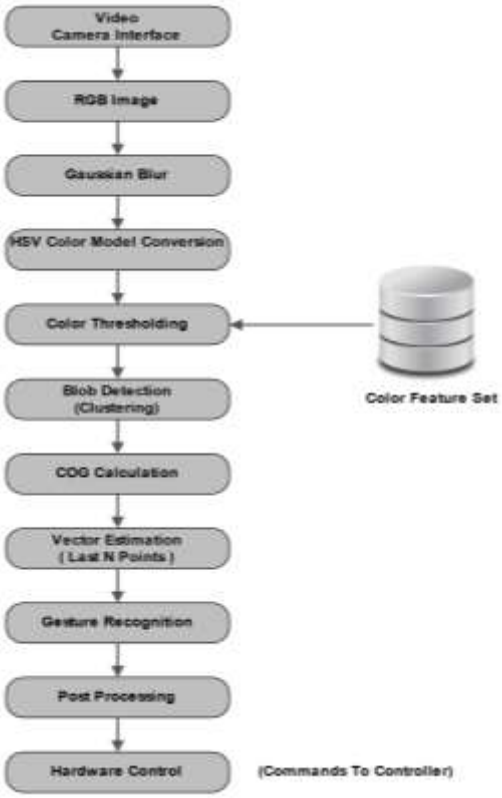

Figure 3: System flow

4. Design & Implementation

4.1 Design

The system design consists of a webcam, set of IC's, an IR sensor and LED and MAX232 cable.

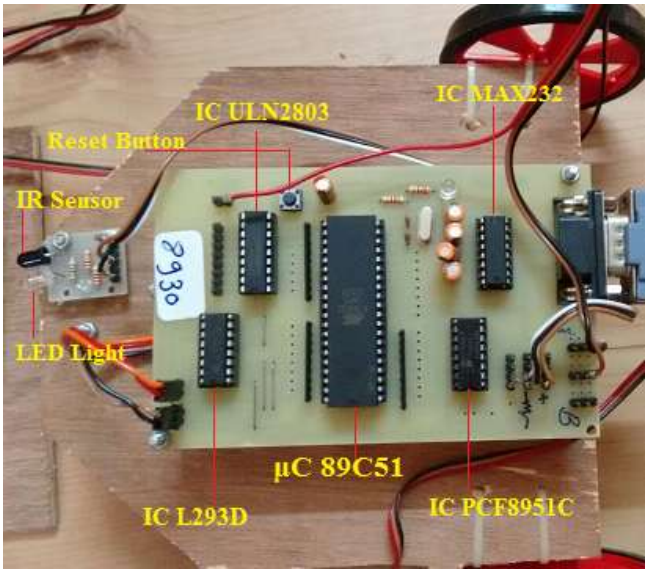

Figure 4: System Design

The microcontroller used in the system is from the 8051 family, i.e. 89C51 IC used to control all the other ICs and the devices connected to the system. 4 IC are connected to the microcontroller 89C51. The connected ICs are:

1. IC ULN2803
2. IC L293D
3. IC PCF8951C

4. IC MAX232

IC ULN2803 is used for device driver, i.e. it's used to connect and control other devices like IR sensors, LED lights. IC L293D is used to control the motors, here 2 motors are connected to the rear wheels of the car buggy. IC PCF8951C is used for logic conversion for microcontroller 89C51 to MAX232. IC MAX232 is used for serial communication between the system and the car buggy. IR sensor has been placed on the front of the car to detect any obstructions in the way simultaneously a LED light is also connected next to IR sensor , that will show that an obstruction has come in its path.

4.2 Implementation

The application is implemented in a very simple manner, the image is firstly captured using a web-cam or any other camera connected to the system to catch the live feed of the hand gestures. Each captured image is processed with a series of algorithm for noise reduction and image processing. The important algorithms processed over the image are Image Blurring, RGB to HSV Conversion, HSV thresholding & Blob detection.

a) Image Blurring

Blur is applied to an image to remove the unwanted noise from the image. The blurring processing firstly includes the separation of Red, Green, Blue components of each pixel of the image. RGB components are separated by using following formulae's

B = (col) AND 0xFF
G = (col >> 8) AND 0xFF
R = (col >> 16) AND 0xFF

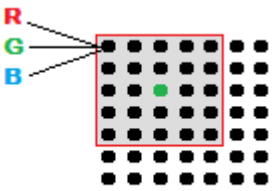

Figure 5: 5x5 matrix

Inorder to choose the pixel to be blurred, a suitable window size is chosen. The window size could be 3x3, 5x5, and 7x7. More the window size better the result. The next step after separating the R,G,B values is that all the r-values, g-values and b-values are added to get new r, g, b namely Rsum, Gsum and Bsum. The Rsum, Gsum and Bsum value is divided by the number of pixels within the window size. For example, if the window size is 3x3, the values are divided by 9.

$$Rsum = \frac{r1+r2+.....+r9}{9}$$
$$Gsum = \frac{g1+g2+.....+g9}{9}$$
$$Bsum = \frac{b1+b2+.....+b9}{9}$$

Finally the new values are put back into a single pixel located at the center of the window.

col = (Rsum | Gsum << 8 | Bsum << 16)

b) RGB to HSV Conversion

This step in the process is very important, as the RGB color model doesn't support natural colors. Whereas the HSV color model does support the natural color. This is required as the webcam fetches live feed that consists of natural colors. Therefore providing us with better results, as this model detects natural colors more effectively than RGB model.

H = Hue (Color)
S = Saturation (Shade)
V = Value (Light)

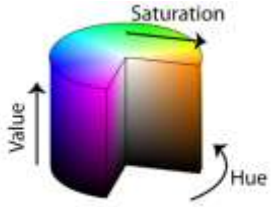

Figure 6: HSV Color Model

This process is taken under only after the blurring algorithm, as the blur algorithm provides the separated RGB values of each pixel from the image. All the separated values are used in conversion to HSV model.

Firstly the rgbmax and rgbmin are found, i.e. maximum and the minimum of the separated RGB values are assigned.

Algorithm:

```
rgbmax = max(r, max (b, g))
rgbmin = min(r, min (b, g))

if (v = 0)  then
    h=s=0                    ... no light, no color

else
    s = 255 * (rgbmax – rgbmin) / v;
    if (s =0)  then
        h=0                  ... no shade, no color
    else
        if(rgbmax == r)
            h = 0 + 43 * (g - b) / (rgbmax – rgbmin)
        else if(rgbmax == g)
            h = 85 + 43 * (b - r) / (rgbmax – rgbmin)
        else if(rgbmax == b)
            h = 171 + 43 * (r - g) / (rgbmax - rgbmin)
      if(h < 0)
            h = 255 + h
```

The above algorithm converts the existing image from RGB Color Model to HSV color model.

c) HSV thresholding


Figure 7: Thresholding sample

Thresholding is a process of creating a black and white image from a color image that consists of setting exactly those pixels to white whose value is above a given threshold, setting the other pixels to black. In this paper we making use of colour bands, therefore the user is supposed to click the colour of the band to set the threshold value. The area of the image coming that

specific colour is changed either black or white according to choice of thresholding colour. Thus it makes it easier to obtain the required part (blob) of the image. In our case the image grabbed would of the hand with the colour bands. Thus the colour band is chosen to separate the hand from the colour bands. Therefore the portion of the colour band would turn black and the unwanted portion would turn white. The colour can be chosen unanimously. Thus giving us a binary image that only consists of the thresholded image, i.e. only the selective part that we require.

d) Blob detection

Blob is nothing but the thresholded image's output of the desired region. Blob detection is a method used to detect specific area/ region in an image. The value of the pixel after applying thresholding would be either black or white.

Figure 8: Detected Blob

Blob detection is a process following image thresholding. Blob detection is done by finding a connected region in an image. The blob is detection detected using simple subtraction. The value of two neighbouring pixel is found out. The difference between the two pixels is calculated. If the horizontal and vertical difference is 1 they belong to the same blob. Thus a connected set of pixels is found leading in detection of the blob, i.e. one of the colour band on the hand.

5. Experimental Results

The system works on the principle of screen division. The screen is divided into three region, based on the finger positions, respective gesture is recognized and appropriate action is performed. The basic actions that the systems performs is shown in the below figure.

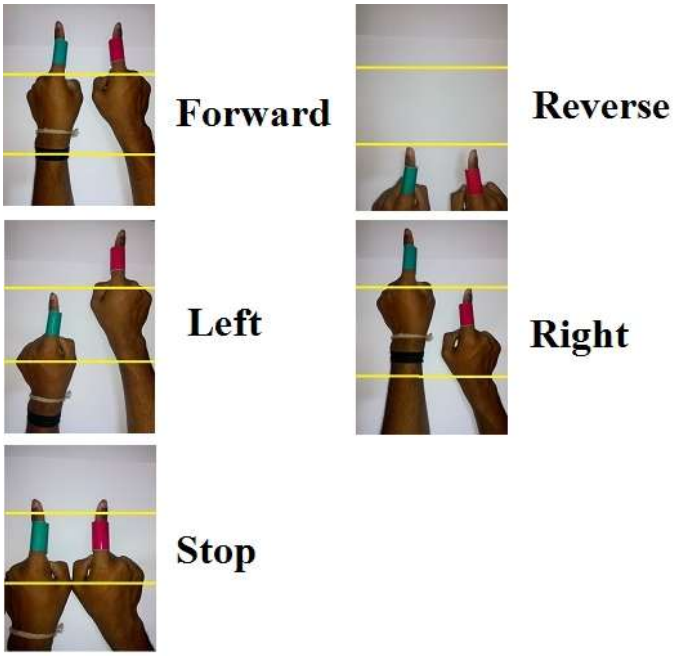

Figure 9: Gestures used in the system

The below table describes, the action being performed on the motors. The motor is capable of moving in clockwise or anti-clockwise motion. Based on various combinations the below actions are possible.

Motor 1	Motor 2	Action Description
Clockwise	Clockwise	Forward
Anti-Clockwise	Anti-Clockwise	Reverse
Stop	Clockwise	Left
Clockwise	Stop	Right
Stop	Stop	Break

Table 1: Working of Motors

The buggy is moved using a set of two motors. If the first motor is set in clockwise motion and the second is also set in the clockwise motion than the action results in forward motion of the buggy. If the first motor is set in anti-clockwise motion and the second is also set in the anti-clockwise motion than the action results in reverse motion of the buggy. If the first motor is set in clockwise motion and the second is stopped than the action results in right motion of the buggy. If the first motor is stopped and the second is set in the clockwise motion than the action results in left motion of the buggy. If the first motor is stopped and the second is also stopped than the action results in no motion of the buggy.

6. Conclusion

A new approach towards hand gestures has been proposed in this paper. We have implemented this system using very feasible items that are easily available to us, providing more accuracy.

7. Acknowledgment

Here we take an initiative to thank all the people who helped us to complete out project, without whom it would be impossible to complete this project without their guidance. At first we would thank our guide, Prof. Raut Anup H. for their never ending guidance, providing the guideline without any complaints at any time and also suggesting all of the ideas and effort they put into our project.
We would also like to thank Prof. Chaudhari S. B. (HOD Comp.) for his interest & co-operation for the project. Expressing our sincere gratitude to TCOER, Pune, for providing us the facilities like Library & Internet. Also thanking all the people who have helped directly or indirectly in completing our project.

References

1. "Using a NIR Camera for Car Gesture Control", B. Ionesco, V. Suse, C. Gadae, B. Solomon, D. Ionescu, S. Islam & M. Cordea, in preceding IEEE Latin America Transaction Vol. 12, No. 3 May 2014.

2. "Hand Gesture Recognition System" Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh, Mahmoud Alhelou, 2011.

3. "Controlling a Car Buggy Using NIR Camera with Hand Gestures", Manish Bhutada, Meenakshi Raut, Mohammed Esoofally, Sanjyot Agureddy, Trinity College of Engineering & Research, Pune, India.

4. "Gesture recognition based on arm tracking for human-robot interaction", Markos Sigalas, Haris Baltzakis and Panos Trahanias.

5. "A Survey on Hand Gesture Recognition in Context of Soft Computing", Ankit Chaudhary, J. L. Raheja, Karen Das, Sonia Raheja.

6. "Hand Gesture Recognition Based on Karhunen-Loeve Transform" Joyeeta Singha, Karen Das. Assam Don Bosco University, Assam, India.

7. "Real –Time Hand Tracking and Gesture Recognition for Human-Computer Interact", Cristina Manresa, Javier Varona, Ramon Mas and Francisco J. Perales, 2000.

8. "Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way: A Survey", Ankit Chaudhary, J. L. Raheja, Karen Das, Sonia Raheja, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, Feb 2011.