Hand Gesture Recognition using Personal Computer to Control a Robotic Car

Shaumika Rane[1] Srisakhi Sengupta[2] Priyanka Kumari[3] Vishakha Gandhi[4]
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]BVCOEW, Pune University

Abstract— In the field of image processing hand gesture recognition is one of the most interesting and challenging genre. Gesture recognition is basically a non-verbal way of communication. Under this title, we will be basically developing a system which recognizes human hand gestures performed in front of the personal computer and uses this gesture for performing movement of a robotic car placed in the vicinity. In this paper web camera is used, after the grabbing of the image we subject the image to image processing algorithms like RGB TO HSV, blur, and thresh holding, dilation, blob detection, segmentation and finally vector calculation. Gesture is recognized and command is given to the microcontroller which enables the motor to move. The system will use a single, colour camera mounted above a neutral colour desk surface next to the computer. The paper briefly describes the schemes of capturing the image from web camera, image detection, processing the image to recognize the gestures as well as few results. This approach can be implemented in real time system very easily.

Key words: Robotic Car, blob detection, Microcontroller-89C51

I. INTRODUCTION

Hand gesture recognition system can be used for interfacing between computer and human using hand gestures. This work presents the interface between the computer and microcontroller (89C51) with the help of MAX RS 232 which provides serial communication. Hand gestures are performed in front of a simple web camera which is mounted on the personal computer. The user will perform the gesture with bare hand. Dynamic hand motions are recognized with the help of some image processing algorithms such as RGB TO HSV, Blurring, Thresh holding, Dilation, Blob detection, segmentation and vector calculation algorithm and the command is given to the microcontroller. As soon as the microcontroller receives a command, it sends it to the motor (L293D) which ultimately performs movements of the car. An attempt is made to use simple algorithms and to provide more user friendly environment.

II. INTRODUCTION TO HARDWARE USED

A. *Microcontroller-89C51:*

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). It is 40 pin IC.

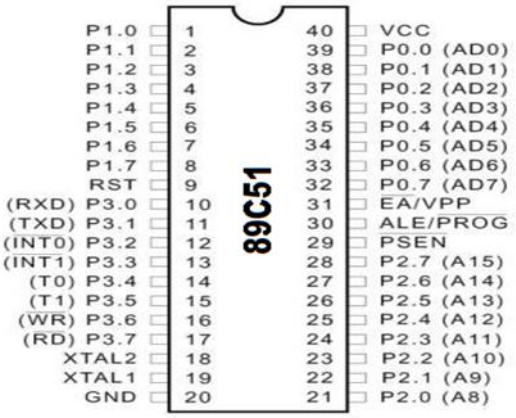

Fig. 1: 89C51 microcontroller pin diagram

1) Specifications:

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Mode

B. *Max -232:*

The MAX232E is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/RS-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/RS-232-F inputs to 5-V TTL/CMOS levels.

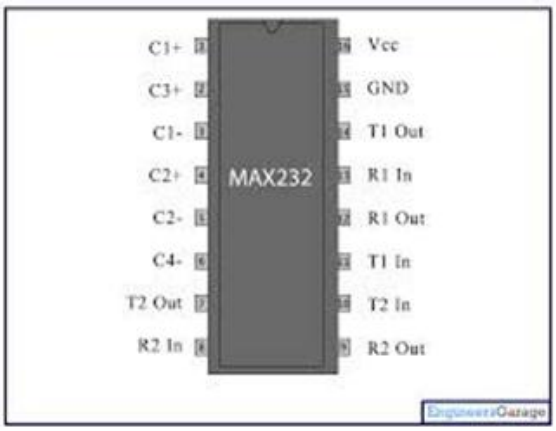

Fig. 2: MAX 232 pin diagram

1) Specifications:

- Meets or Exceeds TIA/RS-232-F and ITU Recommendation V.28
- Operates from a Single 5-V Power Supply with 1.0-µF Charge-Pump Capacitors

- Operates up to 250 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection for RS-232 Bus Pins
- ±15-kV Human-Body Model (HBM)
- ±8-kV IEC61000-4-2, Contact Discharge ±15-kV IEC61000-4-2, Air-Gap Discharge

C. Motor Driver- L293D:

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

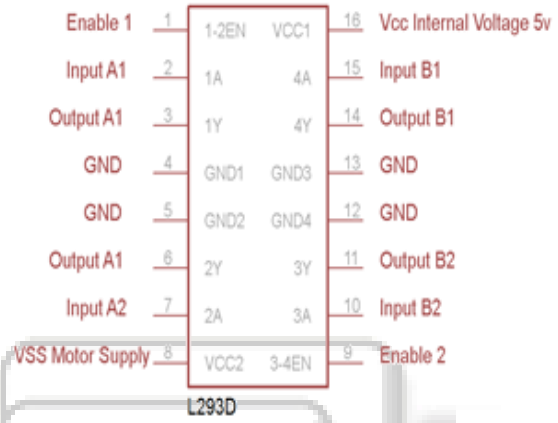

Fig. 3: L293D pin diagram

1) Specifications:

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A per Channel (600 mA for L293D)
- Peak Output Current 2 A per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

III. PROPOSED SYSTEM

This provides details of aims and objectives of the implementation portion and also discusses the methodologies and design principles that were considered while building the system.

The system is basically divided into four modules:

- Image Grabbing[2],[3]
- Image Processing[6],[7]
- Gesture recognition[1],[7],[5]
- Command to microcontroller

A. Image Grabbing:

In order to recognize hand gestures it is first necessary to collect information about the hand from raw data provided by any sensors used. This section deals with the selection of suitable sensors and compares various methods of returning only the data that pertains to the hand. Since the hand is by nature a three dimensional object the first optical data collection method considered was a stereographic multiple camera system. Alternatively, using prior information about the anatomy of the hand it would be possible to garner the same gesture information using either a single camera or multiple two dimensional views provided by several cameras. This system would provide considerably less information about the hand. Some features color ambiguity would be very hard to distinguish since no depth information would be recoverable. Essentially only "silhouette" information could be accurately extracted. The silhouette data would be relatively noise free (given a background sufficiently distinguishable from the hand) and would require considerably less processor time to compute than either multiple camera system. It is possible to detect a large subset of gestures using silhouette information alone and the single camera system is less noisy, expensive and processor hungry. Although the system exhibits more ambiguity than either of the other systems, this disadvantage is more than outweighed by the advantages mentioned above. Therefore, it was decided to use the single camera system. A library called JMyron is to be used to interface Web Camera from within java application. Basically JNI (Java native interface) is used for hardware interfacing in java. After the 24 bit image is grabbed we pass it for image processing.

B. Image Processing:

Image processing is necessary because using intensity alone (black and white) reduces the amount of data to analyze and therefore decreases processor load. Also color also makes differentiating skin from background harder (since black and white data exhibit less variation than color data).

Image processing will involve following algorithms:

- RGB to HSV
- Blurring
- Thresh holding
- Dilation

1) RGB to HSV:

Color vision can be processed using RGB color space or HSV color space. RGB color space describes colors in terms of the amount of red, green, and blue present. HSV color space describes colors in terms of the Hue, Saturation, and Value. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The HSV model describes colors similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors, whereas, HSV describes color using more familiar comparisons such as color, vibrancy and brightness.

- Hue represents the color type. It can be described in terms of an angle on the above circle. Although a circle contains 360 degrees of rotation, the hue value is normalized to a range from 0 to 255, with 0 being red.
- Saturation represents the vibrancy of the color. Its value ranges from 0 to 255. The lower the saturation value, the more gray is present in the color, causing it to appear faded.

- Value represents the brightness of the color. It ranges from 0 to 255, with 0 being completely dark and 255 being fully bright.
- White has an HSV value of 0-255, 0-255, 255. Black has an HSV value of 0-255, 0-255, 0. The dominant description for black and white is the term, value. The hue and saturation level do not make a difference when value is at max or min intensity level.

Converting RGB color to HSV color

```
// r,g,b values are from 0 to 1
// h = [0,360], s = [0,1], v = [0,1]
// if s == 0, then h = -1 (undefined)
void RGBtoHSV( float r, float g, float b, float *h, float *s, float *v )
        {
float min, max, delta;
min = MIN( r, g, b );
max = MAX( r, g, b );
*v = max; // v
delta = max - min;
if( max != 0 )
*s = delta / max; // s
else {
// r = g = b = 0 // s = 0, v is undefined
*s = 0;
*h = -1;
return;
}
if( r == max )
*h = ( g - b ) / delta; // between yellow &
magenta
else if( g == max )
*h = 2 + ( b - r ) / delta; // between cyan & yellow
else
*h = 4 + ( r - g ) / delta; // between magenta & cyan
*h *= 60 // degrees
if( *h < 0 )
*h += 360;
}
```

2) Blur:

There are various ways of implementing the image blurring technique:

1) Linear blur – horizontal or vertical averaging of a fixed number of pixels.
2) Block blur – averaging a small block of pixels by propagating a fixed sized window through the entire image.
3) Gaussian blur – convolution of the image with a two-dimensional Gaussian function.

Image identical to the input image (i.e. the outer part of the image remains not blurred).

1) Traverse through entire input image array.
2) Read individual pixel color value (24-bit).
3) Split the color value into individual R, G and B 8-bit values.
4) Calculate the RGB average of surrounding pixels and assign this average value to it.
5) Repeat the above step for each pixel.
6) Store the new value at same location in output image.

3) Thresh Holding:

Thresh holding is the simplest method of image segmentation. The basic algorithm behind thresh holding is to select those pixel values whose value is greater than threshold (fixed by us) and reject the rest of them.

- P(x,y) =1 if P(x,y)>threshold
  =0 otherwise

Since the selected regions are indicated by 1 and rest by 0 what we get is a binary image.

Thresh holding can be directly applied on RGB model but it is very much dependent upon lighting conditions which is unfavorable in machine vision applications. So we have to switch to better color space such as HSV.

4) Dilation:

Morphological operations are affecting the form, structure or shape of an object. Applied on binary images (black & white images – Images with only 2 colours: black and white). They are used in pre or post processing (filtering, thinning, and pruning) or for getting a representation or description of the shape of objects/regions. The two principal morphological operations are dilation and erosion. Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. Erosion shrinks objects by etching away (eroding) their boundaries. These operations can be customized for an application by the proper selection of the structuring element, which determines exactly how the objects will be dilated or eroded. The *dilation* process is performed by laying the structuring element B on the image A and sliding it across the image in a manner similar to convolution. The difference is in the operation performed. It is best described in a sequence of steps:

- If the origin of the structuring element coincides with a 'white' pixel in the image, there is no change; move to the next pixel.
- If the origin of the structuring element coincides with a 'black' in the image, make black all pixels from the image covered by the structuring element.

Notation:

$$A \oplus B$$

C. Gesture Recognition and Command to Microcontroller:

For gesture recognition we perform the following steps:

We monitor the motion of the finger within a predefined oval. We calculate the deviation of angle between the fingers that are detected. The sine and the cosine of the angle are calculated which are further used to draw lines that are used for vector calculation. We use a variable say currStat to store the value for the gesture recognised. Initially it is set to zero. If a single finger vector is recognised, then we check further angle of deviation of the finger. If the angle is greater than 150 and less than 210, the currStat=1. If angle is less than or equal to 150, then currStat=2. If angle is greater than or equal to 210, then currStat=3 and if the finger vector recognised is 2, then currStat=4.Now, we send command to the microcontroller by writing on its 33rd port of the microcontroller. The commands are in the format (Forward Reverse Left Right).Following commands are given:

- If currStat=1 then 7(0111) for moving forward
- If currStat=2 then 5(0101) for moving left
- If currStat=3 then 6(0110) for moving right

– If currStat=4 then 11(1011) for moving reverse

Depending on the command, the car moves in the desired direction.

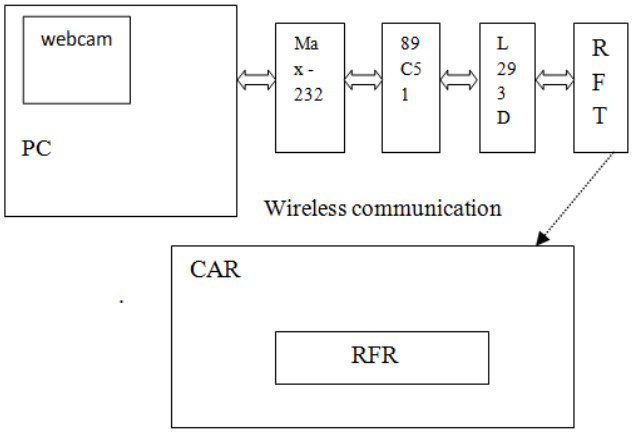

Fig. 5: System architecture

IV. RESULTS

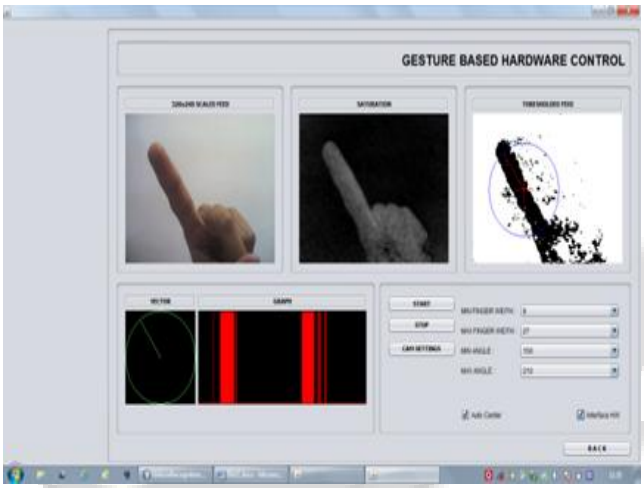

Fig. 4: Gesture Recognition

Gesture is given with bare hand in front of the web camera. Image processing algorithms are performed on the dynamic image captured by the web camera. Vector calculation is done, after which the gesture is recognized and command is given to the microcontroller. With the help of motor the car moves.

This experiment must be carried out in sufficient light. The number of hand gestures can also be added to the system.

V. CONCLUSION AND FUTURE SCOPE

In this paper, a natural and ergonomic hand gesture system is developed for the intelligent Human Computer Interaction. The system is robust to strong illumination changes and can run in real-time for practical purposes. The lexicon size in the designed vocabulary is 4 which cover the comprehensive manipulating commands for robotic car control. The hand gesture vocabulary in the system has many characteristics including high learning ability, low mental load, high comfort, and intuitiveness. Experimental results have demonstrated the efficiency and accuracy of the system which offers available and convenient interface for people and has wide applications in fields of gaming, medical assistance etc. In this paper, adaptive skin color model has been represented. There are four dynamic hand gestures which are one finger up for forward movement, bending one finger towards right for right turn, bending one finger left for turning car left and two fingers up for reverse. All these hand gestures are natural and simple. Though computer vision has been developed since a long time, the commercial applications of computer vision are still few.

At present, a considerable number of research institutes and companies are active in this field. For example, Microsoft's Natal project 2 is developing a new generation of game consoles Xbox 360 which could be operated by hand gestures. To recover the depth information, a two-lens stereo vision camera is deployed to capture the player's movements and postures without requiring user to hold any devices or sensors. Therefore, multi-cameras based approach to investigate more information for recognition of more complicated hand gestures is necessary in the future.

REFERENCES

[1] Ankita Saxena, Deepak Kumar Jain, Ananya Singhal,"Hand Gesture Recognition using an Android Device" 2014 Fourth International Conference on Communication Systems and Network Technologies.

[2] Asanterabi Malima, Erol Özgür, and Müjdat Çetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control"

[3] Ram Rajesh J Nagarjunan D Arunachalam RM and Aarthi R, "Distance Transform Based Hand Gestures Recognition For Powerpoint Presentation Navigation", Advanced Computing: An International Journal (ACIJ), Vol.3, No.3, May 2012

[4] Chen-Chiung Hsieh and Dung-Hua Liou David Lee," A Real Time Hand Gesture Recognition System Using Motion History Image", 2010 2nd International Conference on Signal Processing Systems (ICSPS)

[5] Zhou Ren, Junsong Yuan,Jingjing Meng, and Zhengyou Zhang," Robust Part-Based Hand Gesture Recognition Using Kinect Sensor", IEEE Transactions On Multimedia, VOL. 15, NO. 5, August 2013

[6] Alberto Martin and Sabri Tosunoglu," Image Processing Techniques For Machine Vision"

[7] "Digital Image Processing" by S. Jayaraman, S. Esakkirajan And T.Veerakumar, August 10, 2013, ISBN: 978-0-07-014479-8

