



VIRTUAL CONTROL HAND GESTURE RECOGNITION SYSTEM USING RASPBERRY PI

Geraldine Shirley N.¹ and S. Jayanthi²

¹Embedded Systems Technologies, Sri Ramakrishna Engineering College, Coimbatore, India

²Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore, India

E-Mail: eraldine.shirley31@gmail.com

ABSTRACT

Hand gesture recognition embedded system can be used as an interfacing medium between the computer and human using different hand gestures in order to control the computer. In this proposed system, a real time vision based hand gesture interaction prototype which depends upon finger gestures using color markers is designed. The objective is to develop an embedded system by which one can communicate with any digital device with less hardware requirements and using an external camera to capture the gestures. To avoid the limitations of PC an embedded system consisting of Raspberry Pi which is a Linux based platform is used to identify different color markers on the fingers and when the mouse emulation is started, the software tracks those markers using the camera. The main aim is to create a framework with low cost and effective gesture interpretation system which uses computer vision to analyze different sets of gestures or actions done using the human fingers.

Keywords: embedded system, hand gesture recognition, raspberry Pi, computer vision.

1. INTRODUCTION

Gesture recognition is the process of recognizing and interpreting a stream of continuous sequential gesture from the given set of input data. Gestures are non-verbal information which is used to make computers understand human language and develop a user friendly human computer interface. Human gestures are perceived through vision and this paper aims to use computer vision to analyze different sets of gestures using human fingers and interpret them in order to control the system.

Most of the recognition systems are based on PC but the portability of PC is limited by its weight, size and power consumption. The way to avoid the disadvantages in PC is by using an embedded system which is low cost, power efficient. The approach proposed here is simple and cost effective as it requires less hardware to implement and no sensors are required. The system is developed to interpret set of gestures into mouse control instructions.

FingerMouse is a free computer pointing interface which is used as an alternative to the mouse. A vision system constantly monitors the hand and tracks the color markers placed in the fingertips of each finger and the screen cursor is moved using different gestures.

Double click (left click) and right click is performed using different color markers on the fingers. Raspberry Pi is a Linux based platform which uses Python as the main programming language and software development on Linux is easy as it is an open source code development environment. The system consist of a Raspbian camera which continuously monitors and tracks the gestures.

2. RELATED WORKS

Many researchers have proposed numerous methods for hand gesture recognition system. Abhinav has proposed a wearable gestural interface which lets the user to use natural hand gestures to interact with the

information [1]. The main advantage in this is it integrates digital information into the physical world and its objects and uses hand gestures to interact with digital information, supports multi-touch and multi-user interaction. Some drawbacks are the use of color markers and it is not used in 3D gesture tracking. Saikat et al has proposed a gesture interpretation system capable of controlling the computer mouse using thermal camera [2]. The efficiency of the system is minimized and can be dramatically improved by background subtraction but it is designed only with two hand gestures which is a limited input systems. Shiguo et al has proposed an automatic user state recognition model to control the TV system [3]. Reduced power consumption and computational cost and its limitations are ultrasonic sensor is used for detection.

Experimental results has shown that it is an effective method to use Raspberry Pi board to actualize embedded image capturing system [5]. Dynamic gestures which are performed in complex background [4] can be identified by hand gesture recognition system. Thermal cameras [6] can be integrated along with web camera to identify the hand gestures but in addition to it calibration of thermal camera has to be done periodically. A framework is designed which is a low cost yet effective gesture interpretation system. By using a camera and a tiny projector the system can be controlled using hand gestures during presentations.

The gesture recognition system is an embedded system with less power consumption and efficient image capturing system. This system is more advantageous than PC based systems in terms of cost and portability. Raspberry Pi which is a mini computer is an embedded system which is more efficient in controlling a system through gestures. Linux operating system provides many software choices in order to do a specific task which adds additional functionality in choosing Raspberry Pi.



3. GESTURE RECOGNITION EMBEDDED SYSTEM

a) System components

The system is composed of the Raspberry Pi board and the Raspbian camera to capture the video. The Raspberry Pi board is the central module of the whole embedded image capturing and processing system with Broadcom BCM2835 system-on-chip multimedia processor at 700MHz in which CPU core is a 32 bit ARM1176JZF-S RISC processor. Even though the Raspberry Pi is a computer it does not have a hard drive like traditional computers, instead it relies on the SD card for starting up and storing of information. A 16GB SD card is used in this system. The Raspbian camera module is a 5MP CMOS camera with a fixed focus lens that is capable of capturing still images as well as high definition video. The power supply to the board is 5V and is connected via a micro USB connector. The Raspberry Pi board is connected to the Raspbian camera through the dedicated CSI interface. The system block diagram is shown in Figure-1.

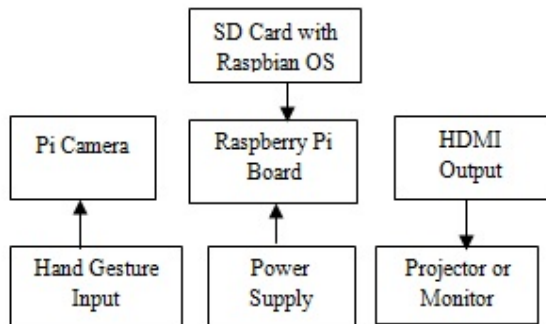


Figure-1. Block diagram of the proposed system.

b) Raspberry Pi

Hardware is a physical device that can be touched or held, like a hard drive or a mobile phone. Software can be thought of as a program or a collection of programs that instruct a computer on what to do and how to do it. Below is an image of the Raspberry Pi which describes some of the components that make up the hardware.

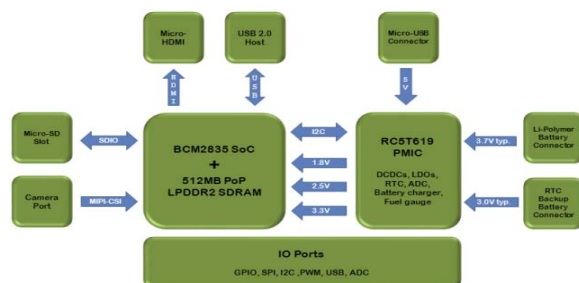


Figure-2. Hardware Components of Raspberry Pi.

The hardware description of Raspberry Pi which is shown in Figure-2 consists of the GPIO pin, SD slot, USB port and Micro USB connector. The GPIO pins are

used for serial communication for interfacing GSM and GPS etc. It uses 16GB SD for installing the Raspbian OS and for storage. The USB port is used for connecting keyboard, mouse, dongle and pen drive. The power supply is given through USB connector.

c) Gesture detection

The system begins by analyzing the captured video frame from the Raspbian camera. Normally the video has to be cut in different images to identify the hand gestures using different color markers. The image obtained is often in the BGR format and it has to be first converted into HSV color space. There are several steps for gesture recognition which is shown in Figure-3.

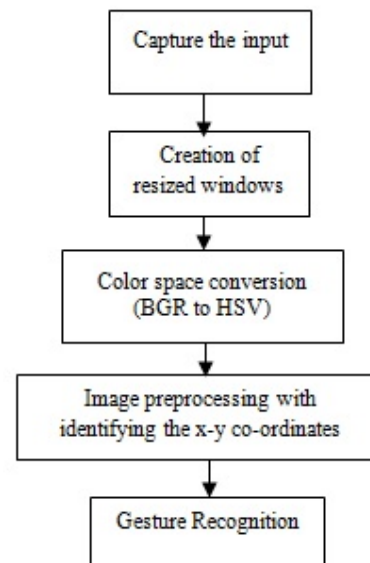


Figure-3. Framework of the system.

Capture the input: The first step is to capture the live video stream from the Raspbian camera. In reference [5], this camera has no infrared filter thus making it perfect to capture images which even during low light conditions. The color markers are identified and the video is converted into picture frames for gesture recognition.

Image acquisition: The next step is to create windows for different color markers which are detected. Resize the frames in order to reduce the resolution which will in turn reduce the computation time. In references [2, 8] noise reduction is done by using the antialiasing filters.

Color space conversion: A proper color model is needed to perform classification. The captured image which is in the BGR color space is converted to HSV (Hue Saturation Value). HSV color space is the most suitable one for color based image segmentation. Hue value represents the shades of the color, S describes how pure the hue color is and V provides the intensity of the color. Different HSV values for different colors are provided to represent the gray image of that particular color. The gesture recognition system is processed dynamically for computation and to reduce the complexity color markers



are used for detection instead of skin tone detection which is used in reference [4]. The resized window which shows the original image detecting the yellow color marker and the converted gray image is shown in Figure-4.



Figure-4. Detection of yellow color marker with the converted gray image.

Image preprocessing: In this phase the color markers are detected and the x-y co-ordinates are identified for the detected image. To identify the x-y co-ordinates of the color markers the size of the monitor is known and the x-y co-ordinates are calculated using pixel co-ordinate system where the position is identified by $px=0$ and $py=0$ corresponding to the top-left corner of the window. For each frame the system recognizes the color markers and the corresponding x-y co-ordinates are displayed. Reference [6] uses background modeling and calibration which makes the system more efficient in computation.

Gesture recognition: To recognize the gestures the gesture can be either one finger or two finger gestures with different color markers. In one finger gesture analysis red color marker is placed in the index finger and it is used to control the cursor movements in the screen. In two finger gesture analysis along with index finger the thumb and middle finger is used for left click and right click. Thus the system is controlled by finger gestures using different color markers. In reference [7] it uses two approaches that are used for cursor movement control which is using resolution mapping and weighted speed cursor control. Using pixel co-ordinate system the resolution size of the window is calculated and mapping is done for cursor movement.

The pseudo code for cursor movement identifying red color marker is shown in Figure-5. It uses the pixel co-ordinate system and the resized window size of the red color marker identification window is used where the HSV value of red color is given and the (mx,my) pixel co-ordinates are calculated. The `getpixel` function is used to identify the red color detected within the resized window and the `move` function is used for cursor movement with respect to the (mx,my) co-ordinates.

```

for x in range(1,299):
for y in range(1,162):
r,g,b=picture.getpixel((x,y))
if rdetect == 0:
if r >= 125 and g <= 60 and b <= 60:
mx=x*4
my=y*3.5
m.move(mx,my)
rdetect = 1

```

Figure-5. Pseudo code for cursor movement.

The pseudo code for mouse left click identifying yellow color marker is shown in Figure-6. Similarly, the same condition is done for identifying the left click where the HSV value changes for yellow color marker. The click function is used to perform the action with respect to the pixel co-ordinates of the window. It has three sub ordinates where the first one represents the maximum size of the x co-ordinate of the window, second represents the maximum size of the y co-ordinate of the window.

```

for x in range(1,167):
for y in range(1,93):
r,g,b=picture.getpixel((x,y))
if ydetect == 0:
if r >= 130 and g <= 70 and b >= 130:
m.click(400,695,1)
m.click(400,695,1)
ydetect = 1

```

Figure-6. Pseudo code for mouse left click.

The pseudo code for mouse right click identifying blue color marker is shown in Figure-7. This is similar to left click where the HSV value for blue color is given and click function takes the pixel co-ordinates as its parameter and performs the right click.

```

for x in range(1,167):
for y in range(1,93):
r,g,b=picture.getpixel((x,y))
if bdetect == 0:
if r <= 60 and g <= 60 and b >= 90:
m.click(mx,my,2)
bdetect = 1

```

Figure-7. Pseudo code for mouse right click.

Using resolution mapping the resolution size of different window screens can be calculated. During presentations the projector screen resolution is mapped with the resized windows of different color markers. These finger gestures can be used to control the next and previous slide buttons during presentation. A smart TV can also be controlled using hand gestures during net access.



4. SOFTWARE IMPLEMENTATION

The Raspberry Pi is a Linux based operating system environment with python as the main programming language. The development environment used here is the IDLE (Integrated Development Environment) which is the basic platform for python. Along with IDLE OpenCV library version 2.4.8 is used for solving computer vision problems. OpenCV-Python is the library of Python bindings designed to solve computer vision problems and it provides all the functions to develop the gesture movements. Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited for many tasks.

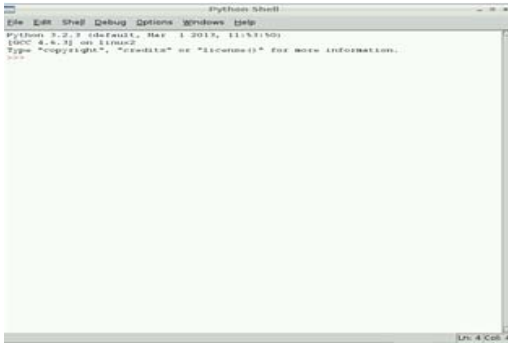


Figure-8. Python shell window.

The Python Shell window shown in Figure-8 is used for running the code and viewing the results of the code. In the menu bar Debug option provides the run module which will debug the code and will show the execution results.

5. EXPERIMENTAL RESULTS

The system consists of Raspberry Pi interfaced with Raspbian camera where the color markers placed in the fingers are used to control the system. The Raspberry Pi interfaced with Raspbian camera along with the monitor window is shown in Figure-9.



Figure-9. Raspberry Pi interfaced with Raspbian camera with the monitor window.

The software running in the background in the Python IDE along with the resized windows for each color detected is shown in Figure-10. It shows the (x,y) co-ordinates for the red color detected which can be used to recognize the cursor movement.

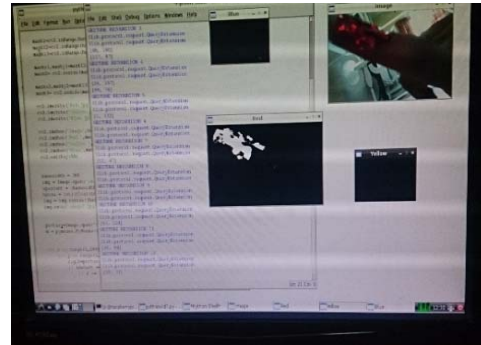


Figure-10. Software running in the background in Python IDE.

The red color marker placed on the index finger is used for cursor movement which is placed on a Desktop icon is shown in Figure-11. The software running at the background is used to identify the red color and the functions are used for cursor movement.



Figure-11. Detection of red color marker for cursor movement.

The blue color marker is used to perform the right click. In Figure-12, the right click is performed showing a pop up menu for the Desktop icon selected using the cursor movement. Once the blue color is detected the software running at the background will perform the right click function on the specified area in the monitor screen.



Figure-12. Detection of blue color marker for right click.



The yellow color marker placed on the finger is used for double click (left click). In Figure-13, the left click is performed showing a pop up menu for the Desktop icon selected. When the yellow color is detected the software running at the background will perform the double click at the pointed area in the monitor.



Figure-13. Detection of yellow color marker for left click.

Thus the system is controlled by finger gestures using the Raspberry Pi which is an efficient and effective embedded system.

6. CONCLUSIONS

This paper provides a computer vision gesture interpretation embedded system which controls the mouse performing different operations for different gestures. Designing an embedded system with Raspberry Pi which is smaller, low cost with less power consumption is more convenient than the traditional PC-based gesture recognition system. The development environment using python language makes it easier to detect the gestures using color markers. In future more gestures can be introduced with different color markers for some functions like scroll, zoom in, zoom out etc. This can be applied in other applications like robotics, interactive video games.

REFERENCES

- [1] Abhinav Sharma, Mukesh Agarwal, Anima Sharma and Sachin Gupta. 2013. "Sixth sense Technology," in Proc. International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 1, pp. 277-282, April.
- [2] Saikat Basak and Arundhuti Chowdhury. 2014. "A Vision Interface Framework for Intuitive Gesture Recognition using Color based Blob Detection," in Proc. International Journal of Computer Applications, Vol. 90, No. 15, March.
- [3] Shiguo Lian, Wei Hu and Kai Wang. 2014. "Automatic User State Recognition for Hand Gesture Based Low-Cost Television Control System," IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, pp. 107-115, February.
- [4] Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh and Mahmoud Alhelou. 2011. "Hand Gesture Recognition System," in Computer Engg Department, The University of Gaza.
- [5] G. Senthilkumar, K. Gopalakrishnan and V. Sathish Kumar. 2014. "Embedded Image Capturing System Using Raspberry Pi System," in Proc. International Journal of Emerging Trends & Technology in Computer Science. Vol. 3, pp. 213-215, April.
- [6] Bobo Zeng, Guijin Wang and Xinggang Lin. 2012 "A Hand Gesture Based Interactive Presentation System Utilizing Heterogeneous Cameras," in Tsinghua Science and Technology. Vol. 17, No. 3, pp. 329-336, June.
- [7] Kamran Niyazi, Vikram Kumar, Swapnil Mahe and Swapnil Vyawahare. 2012. "Mouse Simulation Using Two Coloured Tapes," in Proc. International Journal of Information Sciences and Techniques. Vol. 2, No. 2, pp. 57-63, March.
- [8] P.S. Gotmare. 2013. "Interacting with Computer System using Hand Gesture," in Proc. International Journal of Science, Engineering and Technology Research. Vol. 2, Issue 1, pp. 57-61, January.
- [9] Grigore C. Burdea and Philippe Coiffed. "Virtual Reality ," 2nd Edition Wiley Publications.
- [10] A. Shokoufandeh, I. Marsic and S. Dickinson. 1999. "View-based object recognition using saliency maps" in Image and Vision Computing. Vol 17 No, 5/6, pp. 445-460, April.
- [11] Raspberry Pi [Online]. Available: <http://www.raspberrypi.org>
- [12] Tutorials [Online]. Available: <http://www.raspberrypi.org/tutorial>
- [13] Python [Online]. Available: <https://docs.python.org/2/tutorial>