

A Hand Gesture Recognition Approach towards Shoulder Wearable Computing

Humera Syed¹, M. S. Khatib²
^{1,2} CSE, A.C.E.T/R.T.M.N.U, India

ABSTRACT: Human Computer Interaction needs computer systems and sensing devices. HCI has the limitation of front face interaction with system camera. Since it is not possible to carry out the complete computer system and display system everywhere, therefore there is need of designing and implementing shoulder wearable HCI device. The proposed system can detect hand gestures through short ranged camera. Display of the output will be through mini projection system. Using mini projector we can plot the output almost on any platform. Objectives like visual paint brush application for drawing purpose can be achieved due to which user get more flexibility to draw than a mouse control. This application consists of color selector, brush width, etc. Another objective which can be achieved is developing a call dialing application using GSM modem and GUI application to interact. This application consists of a GSM modem connected to the main processing system and the use of earphone and mike for calling purpose. For achieving these two objectives, two modules, skin detection module and edge detection module is developed initially which contributes to the interactive environment.

Keywords -Wearable Computing, Human Computer Interaction, Hand Gesture Recognition, Skin Detection, Edge Detection.

I. INTRODUCTION

I.1 Hand Gesture Recognition

The basic goal of gesture recognition ^[1] is to have an automated system that can identify the specific human gestures and also use these gestures to control the devices/virtual environment. The gestures that will be performed here are: clicking, double clicking, clicking in hovering state, dragging.

Our primary physical connection to the world is through our hands. We perform most everyday tasks with them. Hand gesture ^{[2][3][4]} is a form of visual communication. The use of hand gestures in man-machine interaction has attracted new interest in recent years. However, when we work with a computer or computer-controlled application; we are constrained by clumsy intermediary devices such as keyboards, mice, and joysticks. These clumsy devices led to the need of using shoulder wearable device. Wearable computing ^[5] using shoulder is a more efficient approach because human head is aligned at a defined angle to the shoulder, such that if head is moved to a particular direction, the shoulder mounted device will also move to the same direction respectively.

I.2 Human Computer Interaction

With the massive influx of computers in society, human computer interaction, or HCI, has become an increasingly important part of our daily lives. It is widely believed that as the computing, communication, and display technologies progress even further, the existing HCI techniques may become a bottleneck in the effective utilization of the available information flow. For example, the most popular mode of HCI is based on simple mechanical devices—keyboards and mice. These devices have grown to be familiar but inherently limit the speed and naturalness with which we can interact with the computer. Thus in recent years there has been a tremendous push in research toward novel devices and techniques that will address this HCI bottleneck. In a computer controlled environment one wants to use the human hand to perform tasks that mimic both the natural use of the hand as a manipulator, and its use in human-machine communication (control of computer/ machine functions through gestures) ^[6].

To exploit the use of gestures in HCI^[7] it is necessary to provide the means by which they can be interpreted by computers. Use of shoulder wearable HCI device helped in removing the bottleneck caused by mouse and keyboard.

II. RELATED WORK

At present the typical main products in multi-touch control technology industry are Diamond Touch, FTIR-Touch. Diamond Touch^[8] is a HW/SW platform that support multi-user concurrent input and gesture interaction, which developed by Mitsubishi Electric Research Laboratories in 2004. It uses electricity induction principle and mounts large number of antennae under the touch panel, where each antenna passes a specific signal. By using the conductivity of the user's own and through his chair, the signal can be transmitted to a separate receiver that each user has. When the user touches the panel, there is a minimum amount of signal transmission among the antennae near the contact point, the user's body and the receiver. The platform has lower touch accuracy, its signal transfer mode restricts the scope of user activities and screen display area, and it has significant limitations point, the user's body and the receiver. The platform has lower touch accuracy, its signal transfer mode restricts the scope of user activities and screen display area, and it has significant limitations.

FTIR (Frustrated Total Internal Reflection) Touch^[9] is a multi-touch hardware platform designed by New York University in 2006, it adopts the frustrated total internal reflection techniques, uneven surface of the fingers causes light beam scattering, scattering light reaches a photoelectric sensor through touch screen, the sensor converts light signals into electrical signals, then the system can obtain the corresponding touch information^[10]. The platform does not require a closed box and can obtain the contact with high contrast. But it requires high performance hardware (LED and compatibility layer), and does not recognize the objects marked. It can only be used in projection display systems today.

III. PROPOSED SYSTEM

Microsoft Surface is a smart desktop compared to FTIR and Diamond touch that supports multi-touch and gesture input, introduced by Microsoft. It adopts image processing technology to implement Multi-touch.

Here a shoulder wearable HCI device is presented. This shoulder-worn implementation allows users to work on interfaces projected onto the environment (e.g., walls, tables), held objects (e.g., notepads, books). On such surfaces - without any calibration - MultiTouch provides capabilities similar to that of a mouse or touchscreen: X and Y location in 2D interfaces and whether fingers are "clicked" or hovering, enabling a wide variety of interactions. Hand gestures are detected by a short ranged camera. In a system there can be more than one cameras but at a time a system can show only one camera. Camera is connected to the camera driver. Each camera has a camera driver. These camera drivers are connected to the operating system. The operating system gives these drivers a unique driver id. Applications contact the operating system for having the camera access.

The camera view consists of live view. Since the camera view is difficult to access as it is handled by the operating system, the live camera view should be converted into frames. The function "hdcTpPicture" will convert the view into frames. Dm Bm is a variable which can hold the entire image in memory. The function "getobject" will load a picture from picture box to bm variable. Even when we get the frame it is not easy to identify the color from it so we need to perform image processing here and as we know that image consists of pixels and each pixel is made up of 3 bits of RGB values so we try to extract RGB value for each pixel and we also try to compare those values with predefined value like what we do in pattern matching. If we get the desired result we process for drawing and all this process will be executed for next frame until we reach the last frame. Here we are doing frame capturing and frame processing both, but it should be compulsory that both the processes should be synchronous for smooth performance.

Using "getBitmapBits" function we can get the pixel RGB value detail which in range of 0 to 255. Here an idata 3D array is provided to hold the pixel details. After execution of this function, the

array filled with pixel detail is achieved where to get the R value of any pixel, $R=idata(2,x,y)$ is used where X, Y are the coordinates of the pixel and same for G and B values. Finally the click event is detected and the action takes place in the form of output. Display of the output will be shown through the mini projection system. Using mini projector the output can be plotted almost on any platform. When the click event takes place, execution of the two modules of Hand Gesture Recognition, that is, Skin Detection module and Edge Detection module takes place for the system to recognize the skin when the touch on the screen takes place.

Traditional Graphical User Interface (GUI) WIMP (windows, icons, menus, pointing device) is the current main human-computer interaction mode. In this interactive mode, the mouse is the primary means of computer operations. But the mouse is only an input device with only 2 degrees of freedom input device, therefore it is hard for people to fully apply the hand operating skills learned in their natural life to human-computer interaction to reduce cognitive burden of the interaction, and improve the efficiency of computer operations. Multi-touch equipments allow one or more than one user to use multiple fingers to interact with computers through graphical user interfaces. Our fingers are of a very high degree of freedom (with 23 degrees of freedom), and can touch directly without any media, which greatly enhances the efficiency of our interaction with computers.

First objective is to provide visual paint brush application for drawing purpose. Due to this user get more flexibility to draw than mouse control. Paint brush application has a color panel as the color selector, brush width etc. Second objective is to develop call dialing application using GSM modem and GUI application to interact. GSM modem will be connected to the main processing system that is the PC system. Calling requires a earphone and a mike for conversation purpose. To achieve these objectives, the two initial modules, Skin Detection module and Edge Detection module are successfully developed.

IV. METHODOLOGY

Camera shows the live view. In this live view there can be any number of objects. Now, suppose a person is standing in a room, this room has various objects like a background, table, chair, ground and many more objects into consideration. Skin recognition module forms a cluster wherever it recognizes the skin pixels. Now, the skin color can match the background color, table color, chair color or it can also match the ground color. Now, wherever the match is recognized, clusters are formed. A cluster is formed when the skin color matches for example to the background color. Likewise various clusters will be formed for various matches of skin color.

In skin recognition we are considering maximum pixels groups having the skin color being detected, so clusters are used to distinguish between the different groups of matches of skin color.

To identify the clusters we need to compare the pixels. Camera can return different values for the same color which can be called as shade of the particular color. But even in the shade, values are in particular range. This range is called as threshold value. For example, camera can return a florescent green shade or a bottle green shade for a typical green color based on the brightness of the light. These shades can have ranges defined which is called as threshold values. Toggle variable is a true or false variable. As we get the skin pixel this variable is set to true and the counter is incremented by one. As we stop getting the pixel, this toggle variable set to false and the counter is set to zero.

As we get the first pixel, we get the starting point $(x1,y1)$, which is the co-ordinates of first pixel identified by cluster. And as we receive the last pixel, we get the end point $(x2,y2)$ which is the co-ordinates of the last pixel identified by cluster. Get nearby value(A,B) function is used where A is the default value of the skin shade and B is the current value to be passed to A. This function checks whether A and B are closer, if yes, then they are the skin shades. Suppose A is assigned with value 50 and its range is defined between 40 to 60. Now we will check If $A+10 > B \ \&\& \ A-10 < B$, if both conditions are true then only A and B are skin shades. If B is 55 then say TRUE, A and B are skin shades. The function `get_skin_pixel(G)` is passed to the `get_nearby_value`. If 300 is the threshold value and If $Count \geq 300$, G is set to TRUE and count is set to $count++$ If $Count < 300$, G is set to FALSE; and count is set to zero.

V. RESULTS AND DISCUSSIONS

The Skin Detection module recognizes only the skin color or the shades of the skin color. Whenever a skin shade is recognized by this module that part of the skin color is set other non skin color part are disabled. The white portion in the below given image is the skin color which is detected by this module and the black portion is the non skin color which is disabled by this module.

In this module we have used a query frame for directly accessing the frame. Along with this grabber is used as an object that grabs the frames. Adaptive Skin Detector is a direct class available in dot net that identifies the skin. Again frame grabber is a function for getting the frames. This module uses Emgu which is a wrapper class of dot net used with Open CV. Open CV is Open Source Computer Vision. It is used as a third party SDK which are used in eye detection, smile detection, skin detection. Software Development Kits are specific, easy to use and understand, accurate and optimized, and it has full hardware support HandGestureRecognition. SkinDetector is used where skin detector is an object. Here the image is taken as the current frame and this current frames' copy is also kept in the image memory. With this the frames' width and height is also taken into consideration. Finally the modified frame is placed into the image box.



Fig.1. Skin Detection Using Hand Gesture Recognition

Finger count detection program using EmguCV technology was developed successfully. Following snapshots shows the identification of count of finger detected by the program and also shows the skin edge detection part. The blue line covering the hand shown in the figure below shows the hull (frame) of the hand displayed and the green line shows the contour (outline) of the hand displayed. The red points represent the outer edge points and the yellow points represent the inner edge points respectively. The figure 1,2,3 shows the count of the figure.

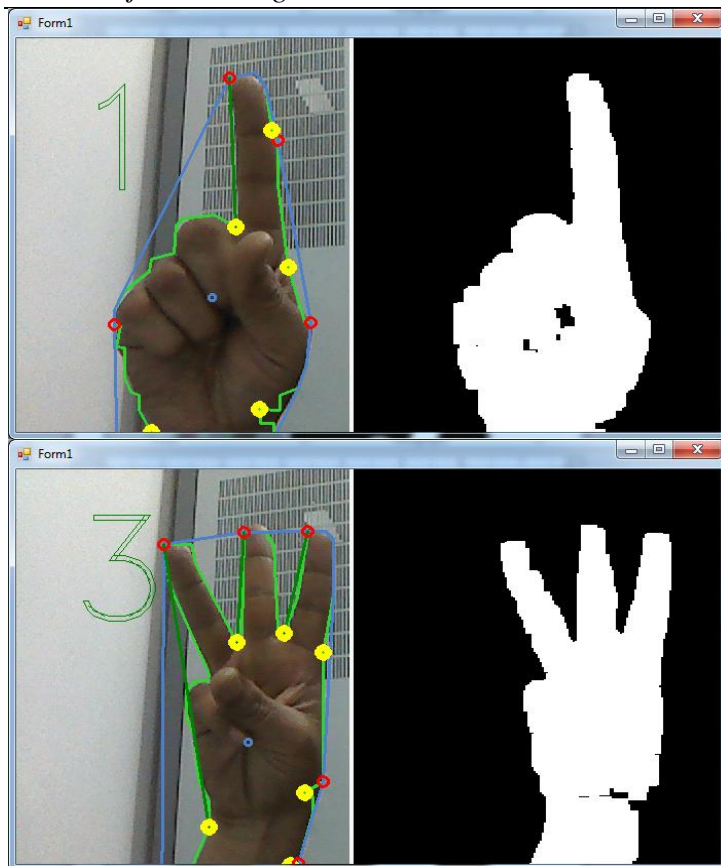


Fig.2. Edge Detection Using Hand Gesture Recognition

VI. CONCLUSION

The execution of the two modules of Hand Gesture Recognition, that is, Skin Detection module and Edge Detection module are developed and executed successfully. These modules will help in interfacing the finger detection module with hardware system for further processing. The finger count detection program successfully displayed the contour and the hull of the hand.

REFERENCES

- [1] Vladimir I. Pavlovic , Rajeev Sharma , Thomas S. Huang , “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review”, VOL. 19, NO. 7, JULY 1997,IEEE.
- [2] Dan Ionescu, “Gesture Control and the New and Intelligent Man-Machine Interface”, 6th IEEE International Symposium on Applied Computational Intelligence and Informatics • May 19–21, 2011,IEEE.
- [3] Sushmita Mitra, “Gesture Recognition A Survey”, VOL. 37, NO. 3, MAY 2007,IEEE.
- [4] Dan Ionescu “Gesture Control and the New and Intelligent Man-Machine Interface”, 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, May 19–21, 2011.
- [5] Carlo Colombo, Alberto Del Bimbo, and Alessandro Valli “Visual Capture and Understanding of Hand Pointing Actions in a 3-D Environment” VOL. 33, NO. 4, AUGUST 2003,IEEE.
- [6] Luigi Gallo and Mario Ciampi, “Wii Remote-enhanced Hand-Computer Interaction for 3D Medical Image Analysis”, 2009,IEEE.
- [7] Deliang Zhu, Zhiqian Feng, Bo Yang, Yan Jiang, Tiantian Yang, “The Design and Implementation of 3D Hand-based Human-Computer Interaction Platform” 2010 International Conference on Computer Application and System Modeling (ICCAISM 2010).
- [8] P. Dietz and D. Leigh, “Diamondtouch: a multi-user touch technology,” in Proceedings of the 14th annual ACM symposium on User interface software and technology. 2001. Orlando, Florida: ACM.
- [9] J. Y. Han, “Low-cost multi-touch sensing through Frustrated Total Internal Reflection,” in Proceedings of the 18th annual ACM symposium on User interface software and technology. 2005. Seattle,WA, USA: ACM.
- [10] C. Pinhanez, et al., “Creating touch-screens anywhere with interactive projected displays,” in Proceedings of the eleventh ACM international conference on Multimedia. 2003, ACM: Berkeley, CA, USA.