Mouse control using Hand Gestures

Ankush Chaudhary[1] Ashish Kumar Sharma[2] Jyoti Dalal[3] Leena Choukiker[4]
[1234]Amity University, Haryana

ABSTRACT- This paper describes research efforts towards new approach for Human Computer Interaction (HCI) and to control the mouse cursor movement and click events of the mouse using hand gestures. The applications of real time hand gesture recognition in the real world are numerous, due to the fact that it can be used almost anywhere where we interact with computers. Hand gesture depend upon camera based color detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual human computer interaction device in a cost effective manner.

Current methods involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, our method is to use a camera and computer vision technology, such as image segmentation and gesture recognition, to control mouse tasks (left and right clicking, double-clicking, and scrolling) and we show how it can perform everything as current mouse devices can.

INTRODUCTION

Human Computer Interaction today greatly emphasizes on developing more spontaneous and natural interfaces. The Graphical User Interface (GUI) on Personal Computers (PCs) is quiet developed, well defined an efficient interface for a user to interact with the computer and access the various applications effortlessly with the help of mice, track pad, etc. In the present day scenario most of the mobile phones are using touch screen technology to interact with the user. But this technology is still not cheap to be used in desktops and laptops. Our objective was to create a virtue mouse system using Web camera to interact with the computer in a more user friendly manner that can be alternative approach for the touch screen.

Touch screen are also a good control interface and nowadays it is used globally in many applications. However, touch screen cannot be applied to desktop systems because of cost and other hardware limitations. By applying vision technology and controlling the mouse by natural hand gestures, reduce the work space required. On developing this approach that uses a video device to control the mouse system. This mouse system can control all mouse tasks, such as clicking (right and left), double clicking and scrolling.

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make Clicking event. One approach, by Erdem et al, used fingertip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region [1, 3]. Another approach was developed by Chu-Feng Lien [4]. He used only the finger-tips to control the mouse cursor and click. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al, used still another method to click. They used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer. Our project was inspired by a paper of Kamran Niyazi [8] et al used Web camera to detect color tapes for cursor movement. The clicking actions were performed by calculating the distance between two colored tapes in the fingers. In their paper [5] K N Shah et al have represented some of the innovative methods of the finger tracking used to interact with a computer system using computer vision. They have divided the approaches used in Human Computer Interaction (HCI) in two categories: 1. HCI without using interface and 2. HCI using interface. Moreover they have mentioned some useful applications using finger tracking through computer vision. This method uses gesture based interactive experiment using finger movements to stimulate mouse operations. As compared with the traditional segmentation method this method has two benefits one is that it uses colored tape and another is that it requires no special object model with relative high performance. These two benefits make the system applicable to the augmented reality systems or other real-time systems. Since the system is based on image capture through a webcam, it is dependent on illumination to a certain extent. Furthermore the presence of other colored objects in the background might cause the system to give an erroneous response. Although by configuring the threshold values and other parameters of the system this problem can be reduced but still that the operating background be light and no bright colored objects be present. The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in a very small amount of time. However a standard pc or laptop has the required computational power for optimum performance of the system. Another fact is that if the resolution of the camera is too high then the system might run slow. However this problem can be solved by reducing the resolution of the image by making changes in the system.

METHODOLOGY

Following are the steps in our approach:

a. Capturing real time video using Web-Camera.
b. Processing the individual image frame.
c. Flipping of each image frame.
d. Conversion of each frame to a grey scale image.
e. Color detection and extraction of the different colors (RGB) from flipped gray scale image.
f. Conversion of the detected image into a binary image.
g. Finding the region of the image and calculating its centroid.
h. Tracking the mouse pointer using the coordinates obtained from the centroid.

Simulating the left click and the right click events of the mouse by assigning different color pointers.

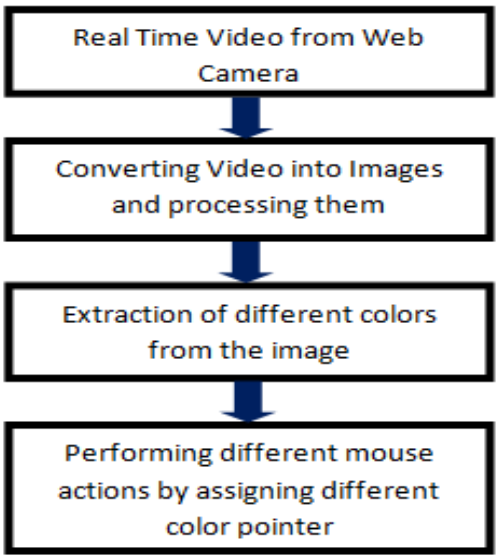

FIG.1. THE BASIC BLOCK DIAGRAM OF THE SYSTEM

A. Capturing the real time video:

For the system to work we need a sensor to detect the hand movements of the user. The webcam of the computer is used as a sensor. The webcam captures the real time video at a fixed frame rate and resolution which is determined by the hardware of the camera. The frame rate and resolution can be changed in the system if required.

- Computer Webcam is used to capture the Real Time Video
- Video is divided into Image frames base on the FPS (Frames per second) of the camera
- Processing of individual Frames

B. Flipping of Images:

When the camera captures an image, it is inverted. This means that if we move the color pointer towards the left, the image of the pointer moves towards the right and vice-versa. It's similar to an image obtained when we stand in front of a mirror (Left is detected as right and right is detected as left). To avoid this problem we need to vertically flip the image. The image captured is an RGB image and flipping actions cannot be directly performed on it. So the individual color channels of the image are separated and then they are flipped individually. After flipping the red, blue and green colored channels individually, they are concatenated and a flipped RGB image is obtained.

C. Conversion of Flipped Image into Gray scale Image:

As compared to a colored image, computational complexity is reduced in a gray scale image. Thus the flipped image is converted into a gray scale image. All the necessary operations were performed after converting the image into gray scale.

D. *Color Detection:*

This is the most important step in the whole process. The red, green and blue color object is detected by subtracting the flipped color suppressed channel from the flipped Gray-Scale Image. This creates an image which contains the detected object as a patch of grey surrounded by black space.

E. Conversion of gray scale Image into Binary scale Image:

The grey region of the image obtained after subtraction needs to be converted to a binary image for finding the region of the detected object. A grayscale image consists of a matrix containing the values of each pixel. The pixel values lay between the ranges 0 to 255 where 0 represents pure black and 255 represents pure white color.

F. Finding Centroid of an object and plotting:

For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An inbuilt function in MATLAB is used to find the centroid of the

detected region. The output of function is a matrix consisting of the X (horizontal) and Y (vertical) coordinates of the centroid. These coordinates change with time as the object moves across the screen.

G. Tracking the Mouse pointer:

Once the coordinates has been determined, the mouse driver is accessed and the coordinates are sent to the cursor. With these coordinates, the cursor places itself in the required position. It is assumed that the object moves continuously, each time a new centroid is determined and for each frame the cursor obtains a new position, thus creating an effect of tracking. So as the user moves his hands across the field of view of the camera, the mouse moves proportionally across the screen.

H. Performing Clicking Actions:

The control actions of the mouse are performed by controlling the flags associated with the mouse buttons. JAVA is used to access these flags. The user has to perform hand gestures in order to create the control actions. Due to the use of color pointers, the computation time required is reduced. Furthermore the system becomes resistant to background noise and low illumination conditions.

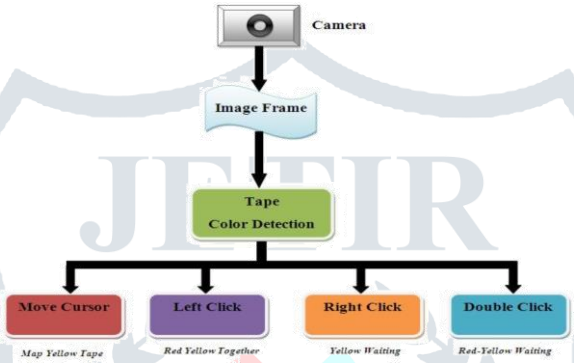

FIG.2. SYSTEM ARCHITECTURE

EXPERIMENT RESULT

A simplification used in this project, which was not found in any recognition methods researched, is the use of a wrist band to remove several degrees of freedom. This enabled three new recognition methods to be devised. The recognition frame rate achieved is comparable to most of the systems in existence (after allowance for processor speed) but the number of different gestures recognized and the recognition accuracy are amongst the best found. Following shows several of the existing gesture recognition systems along with recognition statistics and method.

Paper	Number of gestures recognized	Number of training images	Accuracy
[Bauer & Hienz, 2000]	97	7-hours signing	91.7%
[Starner, Weaver & Pentland, 1998]	40	400 training sentences	96.6%
[Bowden & Sarhadi, 2000]	26	7441 images	97.8%
[Davis & Shah, 1994]	7	10 sequences of 200 frames each	98%
This project	48	90 examples per gesture	98.7%

REFFERENCES

[1]Computer vision based mouse, A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference.

[2]International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 1– Mar 2014

[3]Virtual mouse vision based interface, Robertson P., Laddaga R., Van Kleek M. 2004.

[4]Vision based Men-Machine Interaction http://www.ceng.metu.edu.tr/~vbi/

[5]Chu-Feng Lien, Portable Vision-Based HCI - A Real-time Hand Mouse System on Handheld Devices.

[6]K N. Shah, K R. Rathod and S. J. Agravat, "A survey on Human Computer Interaction Mechanism Using Finger Tracking" International Journal of Computer Trends and Technology, 7(3), 2014, 174-177

[7]James M. Rehg, Takeo Kanade, DigitEyes: Vision-Based Hand Tracking for Human-Computer Interactionin, Proc. of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, Texas, November 1994, pages 16-22.

[8]Gary Bradski, Adrian Kaehler, Learning OpenCV, O'Reilly Media, 2008.

[9]Kamran Niyazi, Vikram Kumar, Swapnil Mahe, Swapnil Vyawahare, "Mouse Simulation Using Two Coloured Tapes", Department of Computer Science, University of Pune, India, International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2, March 2012

[10]J. Serra, Image Analysis and Mathematical Morphology, New York: Academic Press, 1983.

[11]K.Homma and E.-I. Takenaka, "An image processing method for feature extraction of space-occupying lesions," journal of Nuclear Medicine 26 (1985): 1472-1477 Hojoon Park

[12]A Method for Controlling Mouse Movement using a Real-Time Camera,2008.

[13]Stephen Tu, HSV Color Detection Algorithm, White Paper.

[14]Chu-Feng Lien, Portable Vision-Based HCI - A Real-time Hand Mouse System on Handheld Devices.

[15]Shahzad Malik, Real-time Hand Tracking and Finger Tracking for Interaction , CSC2503F Project Report, December 18, 2003

[16]Robertson P., Laddaga R., Van Kleek M., Virtual mouse vision based interface, January 2004

[17]Asanterabi Malima, Erol Ozgur, and Mujdat Cetin, A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control.

[18]Y. Sato, Y. Kobayashi, H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In Proceedings of IEEE, International Conference on Automatic Faceand Gesture Recognition (FG), 2000. pp. 462-467

[19]J. Segen, S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1999. Vol. 1, pp. 479-485.

[20] Hart Lambur, Blake Shaw, Gesture Recognition, CS4731 Project, December 21,2004