

Smart Eye for Computer Application

Swarali Narvekar
EXTC Department
DJ Sanghvi College of Engineering
Vile Parle (W), Mumbai

Manali Godse
Biomedical Department
DJ Sanghvi College of Engineering
Vile Parle (W), Mumbai

ABSTRACT

Despite the advances in computing hardware technologies, the two-button computer mouse has remained the predominant means to interact with a computer. In this paper a system has been designed for implanting an invisible computer mouse that enables interaction with computer without attaching a hardware mouse. Sixth Sense Technology is used for this purpose. The aim of this technique is the use of a real time vision system for its application within visual interaction environments through hand gesture recognition, using general-purpose hardware and low cost sensors, like a simple personal computer or an USB web cam, so that any user could make use of it in his office or home. A fast segmentation process is used to obtain the moving hand from the whole image against different backgrounds and lighting conditions. The movements and arrangements of the hand fingers are grasped as gestures that act as an interaction instruction for the projected application interfaces.

Keywords

SixthSense, Colour Markers, Segmentation

1. INTRODUCTION

'SixthSense' Technology augments the physical world around us with digital information and enables us to use natural hand gestures to interact with that information. By using a camera, 'SixthSense' sees what you see and visually augments any surfaces or objects we are interacting with. It projects information onto surfaces, walls, and physical objects around us, and enables us to connect with the projected information through natural hand gestures, arm movements, or our interaction with the object itself.

2. RELATED WORK

2.1 Mouse Movement using Colour detection

Vision-Based Human-Computer Interaction through Real-Time Hand Tracking and Gesture recognition vision-based interaction is an appealing option used for replacing primitive human computer interaction (HCI) via an invisible mouse. The functions of vlcj to capture the images given by the user and store the captured images in a temporary storage buffer [3]. Each pixel (skipping consecutive four pixels) of the image captured is examined to find the red colour. The pointer reaches the position of the midpoint by examining x and y coordinates of the current area detected. The midpoint of another pixel containing the red colour is found out. The current pixel is compared with previous pixel to move the cursor. The pointer starts moving if the compared two pixels are the same, then. The mouse functions are then performed depending upon the distance detected between the new computed pixels. Another approach works in a continuous manner where camera takes the live video, sending it to the laptop, and MATLAB installed in laptop processes the input and recognizes the colours at the finger tips of the user [1]. The incoming live video from camera is converted into frames

of images obtained from the slicing of video. The images are then processed for colour recognition process. The output of the colour recognition process are the images that contains only those colours of which colour caps are present at the finger tips of the user. For this purpose, RGB values of the colour caps are set prior in the code so that no other colour will be detected in the image after colour recognition except the caps colours. The output is only those colours which are present at the fingertips of the user. The colour is then associated with the mouse cursor in code so that whenever the colour moves in the output image from one position to another, the mouse cursor gets attached at the same position where the colour is now displayed. Mouse Simulation is accomplished using Two coloured Tapes [5]. On the user's hand fingers tapes two different colours are applied. One of the tapes is used for controlling cursor movement and the distance between the two tapes is used for calculating the mouse movement or the cursor movement. The first step in the designed system is to separate out the required hand pixels from the non-hand pixels. Skin segmentation is done after background subtraction. The output of skin segmentation process is an image which is ready for detection of color tapes in the finger. For this an algorithm based on HSV color space is used which is very effective to select a certain color out of an image. The index finger is used to control movement. The system records the distance between the tapes on the index finger and the thumb respectively. Here, the index and thumb must be apart as much as possible so as to get maximum distance. This is the threshold distance for the event which is the cursor movement. The mouse cursor event takes place depending upon the change in the distance between the two coloured tapes.

2.2 Mouse Simulation Using Skin Detection

A system consisting of virtual screen and colour cameras has been developed for colour recognition. The screen with a flat surface is used. The four rectangular coordinates of the same are sent to the system to map it to the workstation screen. The camera is so mounted so that it covers the maximum view. The coordinates are transferred into the system to override the mouse handling operations once the finger comes in contact with the screen. The mechanism operates in three stages. Firstly, a camera captures a real time video of moving hand in front of the screen. The three stages are then performed on the captured frame. In the first stage different implementation models are employed. The colour space model is used to extract the skin pixels; motion detection is used to extract the identified object [4]. The second stage comprises of calculating the probability of the hand identified by convoluting the outputs of different models in stage I to obtain the finger tip. In the third stage, the finger coordinates detected are transferred on the mouse server with the help of sockets [2]. These coordinates are then used to override the system mouse movements. The first stage is the Skin Pixel identification. To identify skin pixels, a colour space model is used. Various colour space models such as RGB, HSV, LUV, YCbCr, etc. have been investigated. The YCbCr component

values of Human Skin colour are calculated. The pixels of the input image are classified into skin colour and non-skin colour clusters by using a thresholding technique that exploits the information of a skin colour distribution map in the YCbCr colour space. Then the object is detection is achieved by building a representation of the scene called the background model or the reference frame. Deviations from the model for each incoming frame are found. Any significant change in an image region from the background model signifies a moving object. Difference in the reference image and the current image is used to find the object. Morphological operations such as Erode, Dilate are performed on the image to enhance the object identification. The edges of the extracted object are identified by canny edge detection algorithm. After processing the edges, different contours are identified. Assuming that the object represents the largest contour, it is extracted in order to detect the finger tip location using structural analysis. The finger tip is located by applying a convex hull algorithm. Each feature either crest or trough belonging to the hand is identified as a defect. The finger tip is then identified from among these defects. The finger tip location gives the position of the coordinates. The coordinates then retrieved are transferred for further processing in the third stage to simulate the mouse movements. The coordinates found through stage II are converted to xml format, so as to transport the coordinates in universal accepted format. These coordinates are fed as an input to the mouse simulator which maps the xml points to mouse coordinates by overriding the mouse handling events. The overridden functions then simulate the mouse movements as per their definitions.

3. SYSTEM DESIGN

The system is designed to operate a computer mouse action by recognizing hand gestures using a simple Crystal web camera. The system is classified into two main modules:

1. Hardware Module
2. Software Module

3.1 Hardware Module

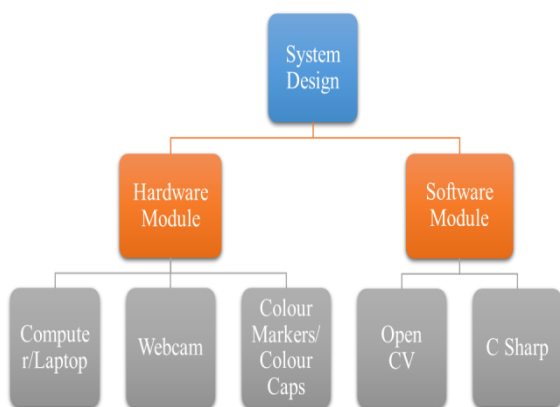


Fig 1: System Design

3.1.1 Computer / Laptop

The system has been designed using Aspire 5742 Acer computer with Intel® Core™ I5-480M at 2.67 GHz of CPU and 2G RAM is used. We have used a 64 bit operating system. The Hard Disk used is 500GB.

3.1.2 Webcam

The in-built Crystal Webcam with 1.3 M pixels resolution is used as the main source of input. It is present on top of the display screen. With a 1.3 megapixel CMOS lens, it is capable of up to a 1280 x 1024 resolution. Its Power Line Frequency

(Anti Flicker) is adjusted to 60Hz. The Acer Crystal Eye webcam captures videos and images with vivid detail and clarity. The first interaction of the real time world takes place with the digital world via a webcam. A webcam captures and recognizes the object in view and tracks the user's hand gestures. The camera, in a sense, acts as a digital eye, seeing what the user sees. It tracks the movements of the thumbs and index fingers of both of the user's hands. It sends the data to the laptop in which Microsoft Visual Studio 2010 and Open CV have been installed for further processing.

3.1.3 Colour Markers

Colour Markers or Colour Caps are used at the tip of the user's fingers. Marking the user's fingers with different colours helps the webcam to recognize gestures. The movements and arrangements of these markers are interpreted into gestures.



Fig 2: Colour Caps

3.2 Software Module

The algorithm had been developed in Open CV. C Sharp programming language has been used for experimental purpose.

3.2.1 Open CV

OpenCV is an image processing library. It contains a large collection of image processing functions. We have installed Microsoft Visual Studio 2010 and downloaded OpenCV Library Version 2.4.2. from their website. The OpenCV library was officially introduced in 1999 by Intel Research initiative to advance CPU-intensive application. The OpenCV library in the earlier version written in C. However since version 2.0, OpenCV includes both C interface and C++ interface. Starting version 2.2, OpenCV can also be built for Android operating system.

3.2.2 C Sharp

C Sharp is a simple, modern, general-purpose, object-oriented programming language. It is full of features that make development faster and easier, usually at the cost of flexibility and/or runtime performance. The execution of the program takes place at a faster rate as compared to the program in C, C++, JAVA and Matlab. An algorithm is developed and implemented using C Sharp for image comparison and motion detection.

4. METHODOLOGY

The objective is to enable mouse movement by detecting the position of user's hand fingers. Fig. 3 shows the algorithm we have used in our approach to move mouse cursor on screen using SixthSense technology. Webcam is used to capture the real time video and send the captured information for further processing. The mechanism works in a continuous manner in which the captured real time information is converted into frames of images. The frames of images are obtained

continuously one after the other and the latest frame is considered further for processing i.e. for the colour recognition purpose. The camera captures the object in a polar fashion i.e. the right side of the real time image appears as left in the captured image and vice-versa. Therefore the captured image is first and foremost resized and flipped so that the captured image is found to be the same as the real time image. The image captured from the webcam is first and foremost resized for processing. It is resized to an image of 320×240 pixels standard value. The size of the screen is (320,240).



Fig 3: Skin Detection

This is present in the program. But the usual monitor size is 1024×768. Therefore we calibrate the screen size to the monitor size. Image is converted from RGB to YCC colour space. The parameters are set at 0, 131, 80 255, 185 and 135 for the detection of human skin. Then Erosion function is used to filter out the smaller noise, but the scope of the necessary information i.e. captured colour marker will also be affected by deformation, so it is necessary to use Dilation function to recover the required information. The sharp contours or edges of the detected object are smoothened by blurring the image by using a Gaussian filter. Threshold being one of the simplest segmentation methods is used to separate out those regions of an image corresponding to the objects which we want to analyze. This separation is based on the variation of intensity between the object pixels and the background pixels. Once the important pixels are separated properly, we have set them with a determined value to identify them i.e. we have set the values in between 20 to 255. The cursor movement is associated with the change in the pixel value from one frame to another. Threshold binary is the function type which is used to perform the threshold operation. The palm area i.e. the contour of the palm is found out by using Contour Approximation function in Open CV. The palm area is calculated using convexity defect. The required set of points is created from the convexity defect depth points with the help of which the centre of the palm is detected using minimum enclosing circle. The next step is the detection of finger points i.e. tip of all the fingers. The palm bounding circle is drawn joining all the finger points. Then the distance between the centre of the palm and the finger point is calculated. This distance keeps on changing from one frame to another as the user moves his/her fingers. In OpenCV it is necessary to import the dll file i.e. dll file user32.dll to control the mouse pointer. The gestures shown in Fig. 4 have been used extensively to perform the Left click and right click operations respectively.

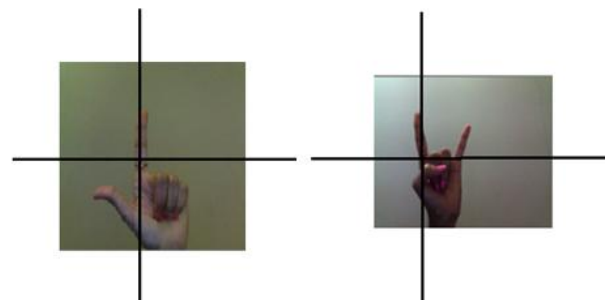


Fig 4: Gestures used for Left click and Right click

5. RESULTS

The click trigger is set at 10 seconds and it indicates that the single click operation would take place if the hand is maintained in the required gesture or posture in between 10 to 20 seconds. The hand is held constant in the above position for 18 seconds and hence the single left click operation takes place as shown in Fig. 4.



Fig 5 (a): Left Single Click Detection



Fig 6(a): Left Double click and Drag Operation Detection

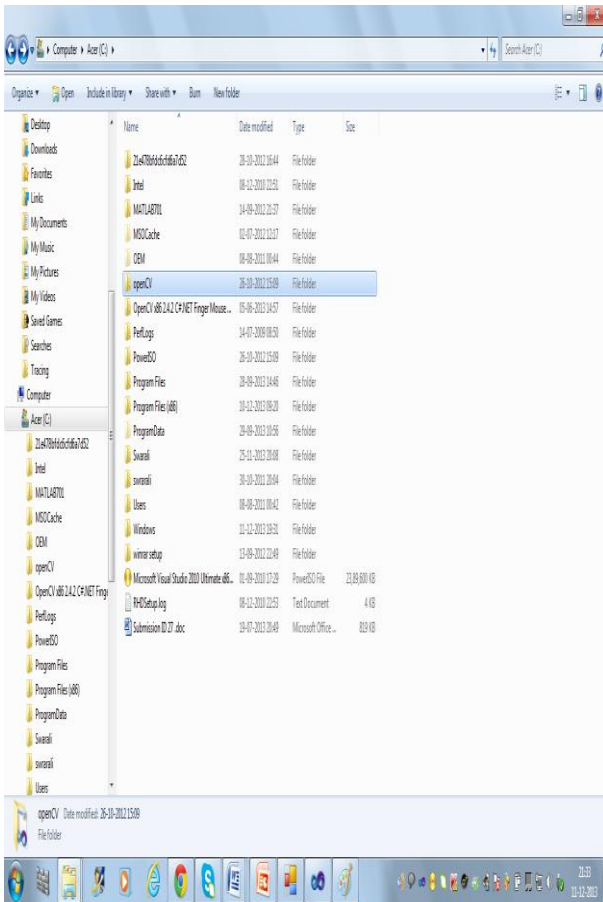


Fig 5(b): Left Single Click

Similarly a left double click operation would take place if the hand is held in the required posture in between 20 to 30 seconds and the drag operation would take place if the hand is held in the required position for more than 30 seconds.

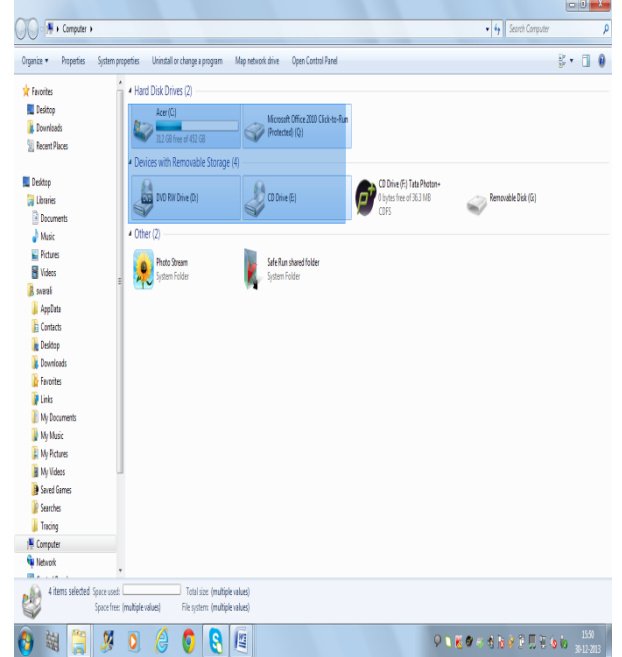


Fig 6(b). Left Double Click and Drag Operation

The timer count is set between 10 to 20 seconds for right single click. The timer count is set in the interval of 20 to 30 seconds to perform the right double click operation. When the hand is maintained in the above posture for about 17 seconds the right single click operation is performed as shown in the below Fig 8.

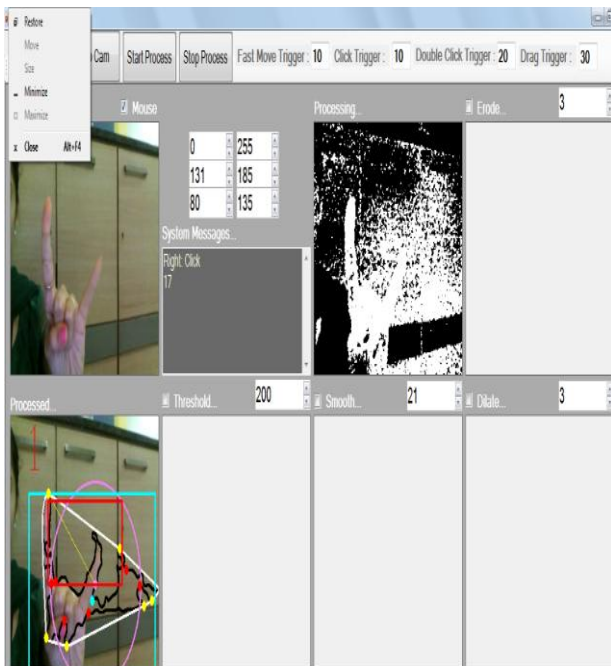


Fig 7. Right Single Click Operation

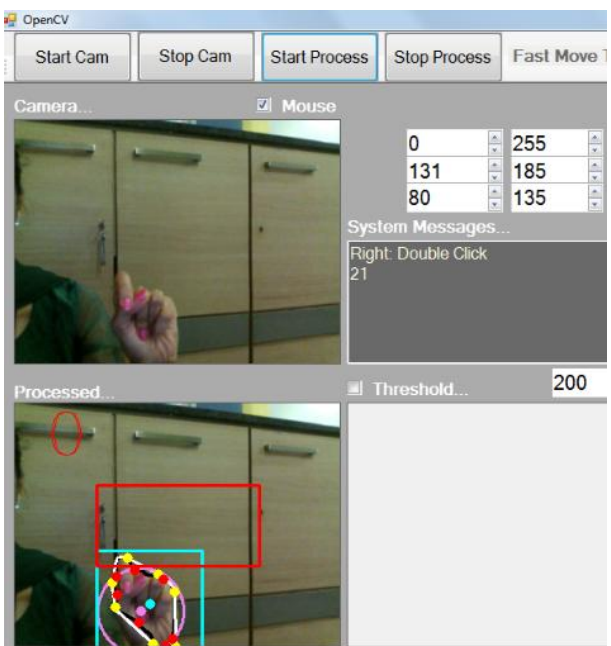


Fig 8. Right Double Click Operation

Each hand gesture was successfully detected and tracked. The above mentioned system can also detect and differentiate between different gestures even if hand is at different angles with respect to horizontal axis successfully to a great extent. Moreover, it can also differentiate gestures when hand is rotated along vertical axis as long as fingers do not overlap.

6. CONCLUSION AND FUTURE WORK

The system based on image recognition and motion detection technology is used to perform mouse control operations. The main focus is to implement a vision based interaction system that uses standard hardware and extends already well known interaction metaphors. The mouse operations are associated

with the skin detection process instead of colour detection process. The use of colour markers is completely eliminated by associating the cursor with the human skin thereby reducing the hardware components. The palm, palm points and the finger points are calculated and the cursor movement is associated with the change in the skin pixel position per frame. The system has undergone informal user testing and from the results it can be seen that the combination of timer and gesture recognition is used to perform different mouse control operations such as scrolling, single click, double click and drag operation. The combination between timer and gesture recognition is used to perform different mouse control operations such as scrolling, single click, double click and drag operations. However, the performance of the system can be increased further by using a more powerful camera and a faster processor for this system. This system could be useful in presentations and to reduce work space.

7. REFERENCES

- [1] Kunal Gupta, Prateek Agrawal, Mouse Movement Through Finger By Image Grabbing Using Sixth Sense Technology, , [IJESAT] International Journal Of Engineering Science & Advanced Technology Volume-2, Issue-2, 245 – 249 Mar-Apr 2012.
- [2] Mridul Birla, Rahul Kulal, Shruti Bardia and Sugandha Rudra TouchX- Mouse Simulation Using Gestures on a Virtual Screen Department of Information Technology, Pune Institute of Computer Technology, Pune, India 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science.
- [3] Priya Philip, T M Bhuruguram Shany Jophin, Sheethal M.S, Gesture Based Interface Using Motion and Image Comparison, International Journal of Advanced Information Technology (JJAIT) Vol. 2, No.3, June 2012.
- [4] Swaminathan P.L. and Vivek Veeriah J. Robust Hand Gesture Recognition Algorithm for Simple Mouse Control International Journal of Computer and Communication Engineering, Vol. 2, No.2, March 2013.
- [5] Kamran Niyazi, Swapnil Mahe, Swapnil Vyawahare and Vikram Kumar, Mouse Simulation Using Two Coloured Tapes International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2, March 2012.
- [6] K.Arai & R. Mardiyanto Camera as Mouse and Keyboard for Handicap Person with Troubleshooting Ability, Recovery, and Complete Mouse Events International Journal of Human Computer Interaction (IJHCI), Volume 1 Issue 3.
- [7] Qinmu Peng and Yiu- Ming Cheung, Eye Gaze tracking with a Web camera in desktop environment, Human-Machine Systems, IEEE Transactions ,Volume 45, Issue 4 July 2015.