JOURNAL OF INFORMATION, KNOWLEDGE AND RESEARCH IN ELECTRONICS AND COMMUNICATION ENGINEERING
ISSN: 0975 – 6779| NOV 16 TO OCT 17 | VOLUME – 04, ISSUE – 02

VIRTUAL MOUSE

[1]TEJASWINI S GALANDE, [2]SNEHAL L JADHAV, [3]REKHA C KAMBLE, [4] RAJSHREE S JAMDAR
Department of Electronics & Telecommunication Engineering,
Shree Chhatrapati Shivajiraje College of Engineering
Dhangawadi, Pune,India.

[1]tejaswinigalande94@gmail.com
[2]jadhavsnehal1111@gmail.com
[3]rekhakambale2@gmail.com
[4]rajashree-shinde@redifmail.com

ABSTRACT :
Abstract*: Since the computer technology continues to grow up, the importance of human computer interaction is enormously increasing. Nowadays most of the mobile devices are using a touch screen technology. However, this technology is still not cheap enough to be used in desktop systems. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen. In this study, finger tracking based a virtual mouse application has been designed and implemented using a regular webcam. The motivation was to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device.*

KEYWORDS *: Enormously, vision technique, implemented.*

IJIKREECE

1. Introduction

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 a 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. Another approach was developed by Chu-Feng Lien. 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."In this study, a color pointer has been used for the object recognition and tracking. Left and the right click events of the mouse have been achieved by detecting the distance between thumb and index finger, thumb and middle finger respectively. The motivation was to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device

Our main objective in this project is to control mouse cursor using software in real time and provide user with the same user experience as that of the hardware mouse with minimum delay. One of the main objective is to reduce the hardware and also decrease the processor consumption (compared to other software's such as MATLAB)

2.SYSTEM DESIGN

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read-only memory (PEROM).

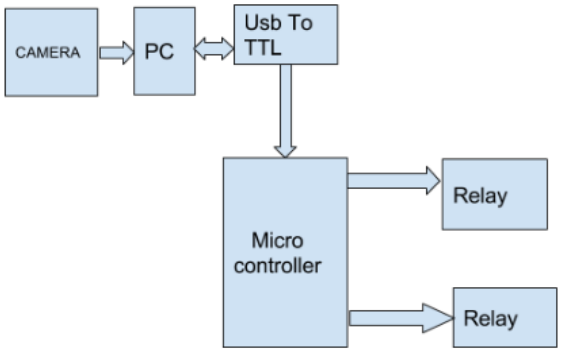

Fig. 1 Block Diagram of proposed

1.1. Image acquisition setup: It consists of a web camera with suitable interface for connecting it to PC.
1.2. Processor: It consists of personal computer or a dedicated image processing unit.

1.3. Image analysis: Certain tools are used to analyze the content in the image captured and derive conclusions e.g. Matlab 7.0

Machine control: After processing, some conclusions has to be made in order to initiate control actions. In our case control actions are desktop control via mouse control.

Colour stickers (R G B)

Fig.2: Colour Stickers

It is at the tip of the user's fingers. Marking the user's fingers with red, green, and Blue tape helps the webcam recognize gestures. The movements and arrangements of these Markers are interpreted into gestures that act as interaction instructions for the projected application interface.

3.SOFTWARE AND HARDWARE REQUIRMENTS:

1) Image processing softwareImage processing toolboxAcquisition processing toolbox
2) MATLAB
3) Webcam(Normal)
4) Computer or laptop
1GB RAM
10GB free hard disk
5) Colour stickers (R G B)

1. It can be projected on any surface or you can type in the plain air.
2. It can be useful in places like operation theatres where low noise is essential.
3. The typing does not require a lot of force. So easing the strain on wrists and hands.
4.The Virtual Keyboard is not restricted to the QWERTY touch-typing paradigm;adjustments can be done to the software to fit other touch-typing paradigms as well.
5. No driver software necessary, it can be used as a plug and play device.
6. Portable.
7. Cost effective.

4. FEATURE

- Paint drawing
- Scrolling of ppt
- Music player control
- Clicking of a pic
- Emergency call
- Emergency sms

5. FLOWCHART

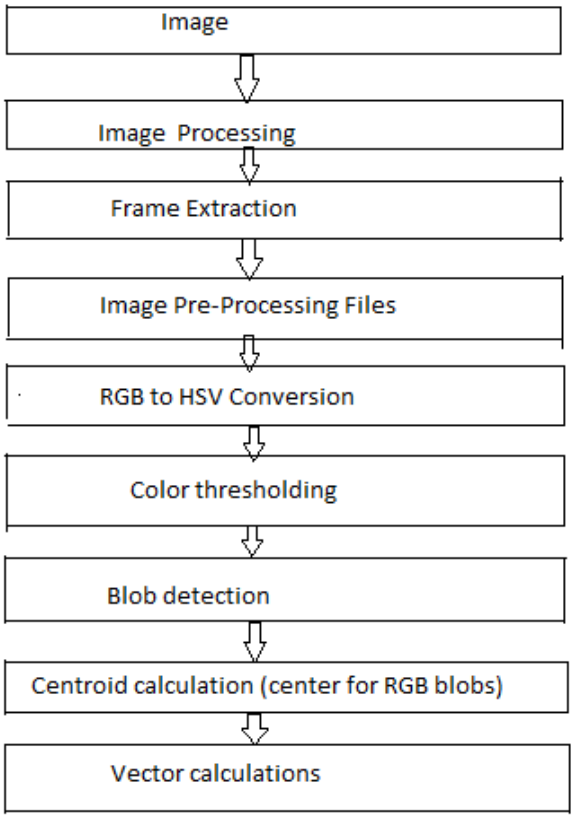

Fig.3:Flowchart

5.1-Image:Images are defined over two dimension digital image processing may be modeled in the form of multidimensional system.

5.2image processing:Image processing is processing of images using mathematical operations by using any form of signal processing.

5.3Frame Extraction: A price to be paid for this extreme flexibility in handling linear and non-linear operations is that a number of anomalies caused by the camera, such as geometric distortion,MTF roll-off, vignetting, and non-uniform intensity response must be taken into account or removed to avoid their interference with the information extraction process.

5.4Image pre-processing files:To improve the image in ways that increase the chances for success of the other processes .

5.5RGB to HSV conversion: $R' = R/255$

$G' = G/255$
$B' = B/255$
$Cmax = \max(R', G', B')$
$Cmin = \min(R', G', B')$
$\Delta = Cmax - Cmin$
Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:
$V = Cmax$

5.6 Color thresholding: The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T (that is, or a white pixel if the image intensity is greater than that constant. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming completely white.

5.7 .Blob detection: blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors: (i) *differential methods*, which are based on derivatives of the function with respect to position, and (ii)*methods based on local extrema*, which are based on finding the local maxima and minima of the function. With the more recent terminology used in the field, these detectors can also be referred to as *interest point operators*, or alternatively interest region operators (see also interest point detection and corner detection).
There are several motivations for studying and developing blob detectors.

5.8.Centroide calculations:
The problem is that I need a simplified version of algorithm, which can calculate centroid of several white contours in a binary image. For example, if there is only one white contour, coordinates Xc and Yc of the contour center are calculated using formula:

$$X_c = \frac{1}{M}\sum_{i=1}^{n} x_i m_i$$

where M is the sum of intense m_i, m_i is the pixel intense value, x_i and y_i are pixel location on the image, n is the total number of pixels.\

5.9Vector calculation: The vector functions operate on three-dimensional vectors, i.e. groups of three numbers (see below). Vectors are used for different purposes, mostly for describing a x y zposition, a direction or speed, or a blue green red color , or a pan tilt roll 3-dimensional Eulerangle.Any var array of length 3, or any three consecutive skills or var parameters of an object can be used as a vector. For instance, to use the my entities' x y z parameters as a position vector, just pass my.x to the vector function. Thus, valid vector parameters are anypredefined VECTOR, ANGLE,
or COLOR struct, and any variable that is defined as var[3];any element of a variable array that is at least 3 less then the length of the array (like array[0], array[3], array[6]...);any entity skill between skill1 and skill96;the x, pan,
or blue parameter of any entity, particle, or view.a vector is characterized by a **magnitude** (length) and a **direction**. A vector consists of several numbers and is defined like this:**V = [e1, e2, e3, ..., en]**

5.RESULT

Fig .3. Virtual Mouse Used as A Virtual Marker

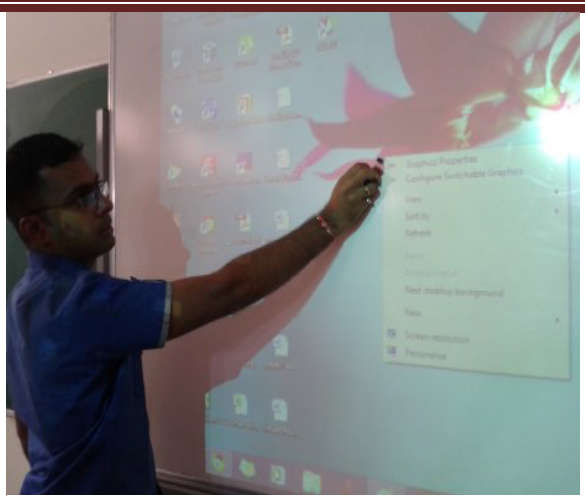

Fig 4. Right Click Using Virtual Mouse 2

The current implementation performs a number of image processing operations on the entire image after every frame capture. While the speed of the tracker is quite good, we could possibly improve the tracking rate significantly by tracking local features of the hand instead (e.g. the fingertips). The downfall with this approach is that moving the hand too quickly could result in tracking failure, but we could always simply revert back to the current implementation to reinitialize the tracker. Finally, the current implementation only uses the 2D finger axis from either the left or right image as a measure of finger orientation It would be desirable to determine the 3D axis of the finger in order to detect finger orientations in the z (depth) direction as well. It turns out that this could be accomplished quite easily.The first plane would pass through the center of projection of the left camera and through the 2D image line for the finger in the left image. Similarly, the second plane would pass through the center of projection of the right camera and through the 2D image line for the finger in the right image

6. .CONCLUSION

This project presented a vision-based hand tracking system that does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras. Specifically, the system can track the tip positions of the thumb and index finger for each hand, assuming that a calibrated pair of cameras is viewing the hands from above with the palms facing downward. The motivation for this hand tracker was a desktop-based two-handed interaction system in which a user can select and manipulate 3D geometry in real-time using natural hand motions. The algorithmic details for the hand tracker were presented, followed by a discussion of the performance and accuracy of the system, as well as a discussion of how the system could be improved in the future would still be required in order to differentiate between other objects with skin-coloured pixels, such as faces.

7. FUTURE SCOPE

This hand tracking system in an augmented reality setting where a user, wearing a head-mount display, could interact with virtual 3D be attached to the head-mount display and viewpoint could thus be controlled by natural head motions, resulting in a changing background scene.

.If the skin pixel detector could be made more robust, it would be possible to completely discard the background subtraction phase and use the current system in such an augmented reality setting. However, a more sophisticated hand segmentation system between these two planes would then represent the axis of the finger in 3D

9. REFERENCES:

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

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

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

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

[5] Hailing Zhou, Lijun Xie, Xuliang Fang, Visual Mouse: SIFT Detection and PCA Recognition, cisw, pp.263-266, 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007),2007

[6] 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.

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

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

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

[10] 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.