

Gesture Based Interface Using Motion and Image Comparison

Shany Jophin¹, Sheethal M.S², Priya Philip³, T M Bhruguram⁴

Dept of Computer Science Adi Shankara Institute of Engineering
And Technology, Kalady,
shanyjophin.s@gmail.com

ABSTRACT

This paper gives a new approach for movement of mouse and implementation of its functions using a real time camera. Here we propose to change the hardware design. Most of the existing technologies mainly depend on changing the mouse parts features like changing the position of tracking ball and adding more buttons. We use a camera, colored substance, image comparison technology and motion detection technology to control mouse movement and implement its functions (right click, left click, scrolling and double click).

KEYWORDS

HCI, Sixth Sense, VLCJ

1. INTRODUCTION

Human computing interaction (HCI) is one of the important area of research where people try to improve the computer technology. Nowadays we find smaller and smaller devices being used to improve technology. Vision-based gesture and object recognition are another area of research. A simple interface like embedded keyboard, folder-keyboard and mini-keyboard already exists in today's market. However, these interfaces need some amount of space to use and cannot be used while moving. Touch screen are also globally used which are good control interface and are being used in many applications. However, touch screens can-not be applied to desktop systems because of cost and other hardware limitations. By applying vision technology, colored substance, image comparison technology and controlling the mouse by natural hand gestures, we can reduce the work space required. In this paper, we propose a novel approach that uses a video device to control the mouse system properly.

2. RELATED WORK

2.1. Mouse free

Vision-Based Human-Computer Interaction through Real-Time Hand Tracking and Gesture Recognition Vision-based interaction is an appealing option for replacing primitive human-computer interaction (HCI) using a mouse or touchpad. We propose a system for using a webcam to track a users hand and recognize gestures to initiate specific interactions. The contributions of our work will be to implement a system for hand tracking and simple gesture recognition in real time [1].

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 finger tip 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 [2, 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.

2.2. A Method for Controlling Mouse Movement using a Real-Time Camera

This is a new approach for controlling mouse movement using a real-time camera. Most existing approaches involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, we propose to change the hardware design. Our method is to use a camera and computer vision technology, such as image segmentation and gesture recognition. 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 current mouse devices can. This paper shows how to build this mouse control system [5].

2.3. Sixth Sense

'SixthSense' is a wearable gestural interface that augments the physical world around us with digital information and lets us use natural hand gestures to interact with that information. The SixthSense prototype is comprised of a pocket projector, a mirror and a camera. The hardware components are coupled in a pendant like mobile wearable device. Both the projector and the camera are connected to the mobile computing device in the users pocket. The projector projects visual information enabling surfaces, walls and physical objects around us to be used as interfaces; while the camera recognizes and tracks user's hand gestures and physical objects using computer-vision based techniques [6].

2.4. Vlcj

The vlcj project is an Open Source project that provides Java bindings for the excellent vlc media player from Video LAN. The bindings can be used to build media player client and server software using Java - everything from simply playing local media files to a full-blown video-on-demand streaming server is possible. vlcj is being used in diverse applications, helping to provide video capabilities to software in use on oceanographic research vessels and bespoke IPTV and home cinema solutions. vlcj is also being used to create software for an Open Source video camera at Elphel and video mapping for the Open Street Map project [7].

2.5. Mouseless

Mouseless is an invisible computer mouse that provides the familiarity of interaction of a physical mouse without actually needing a real hardware mouse. The Mouseless invention removes the requirement of having a physical mouse altogether but still provides the intuitive interaction of a physical mouse that we are familiar with. Mouseless consists of an Infrared (IR) laser beam (with line cap) and an Infrared camera. Both IR laser and IR camera are embedded in the computer. The laser beam module is modified with a line cap and placed such that it creates a plane of IR laser

just above the surface the computer sits on. The user cups their hand, as if a physical mouse was present underneath, and the laser beam lights up the hand which is in contact with the surface. The IR camera detects those bright IR blobs using computer vision. The change in the position and arrangements of these blobs are interpreted as mouse cursor movement and mouse clicks. As the user moves their hand the cursor on screen moves accordingly. When the user taps their index finger, the size of the blob changes and the camera recognizes the intended mouse click.[8]

2.6. Real-time Finger Tracking for Interaction

In this work, they described an approach for human finger motion and gesture detection using two cameras. The target of pointing on a flat monitor or screen is identified using image processing and line intersection. This is accomplished by processing above and side images of the hand. The system is able to track the finger movement without building the 3D model of the hand. Coordinates and movement of the finger in a live video feed can be taken to become the coordinates and movement of the mouse pointer for human-computer interaction purpose.[9]

3. PROPOSED WORK

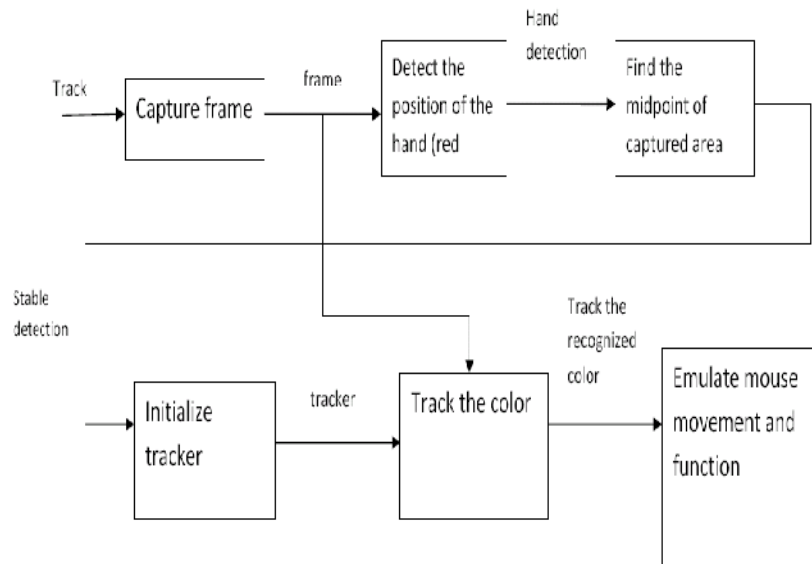


Figure 1. The proposed working

3.1. Product and system features

3.1.1. Computer

(Laptop)-Processor: Pentium4, Processor Speed: 1GHz, RAM Capacity: 512 MB, Hard disk: 40GB, Monitor: 15SVGA.

3.1.2. Webcam

It is used for the image processing and also for capturing the image.

Video data format: 12.24-bit RGB, Image resolution: Max 2560*2084, Software enhanced menu display/sec: 30 in CIF mode, Menu signal bit: 42db, Lens: 6.00mm, Vision: +/-28, Focus range: 3 centimeter to limitless.

3.1.3. Finger Tip

(Red and blue colored substance)- it is used as an alternative for mouse and control the functions of pointer.

3.1.4. Software Requirements

Operating System: Windows XP, Windows vista, Windows 7.
Code Behind: JAVA (Red Hat or Eclipse).

3.1.5. Internal Interface Requirements

Swing, vlcj.

3.2. Tools used

Laptop (include software like red hat, eclipse, vlcj), webcam, a colored device. The performance of the software is made to be improved by examining each pixel leaving behind its four consecutive pixels.

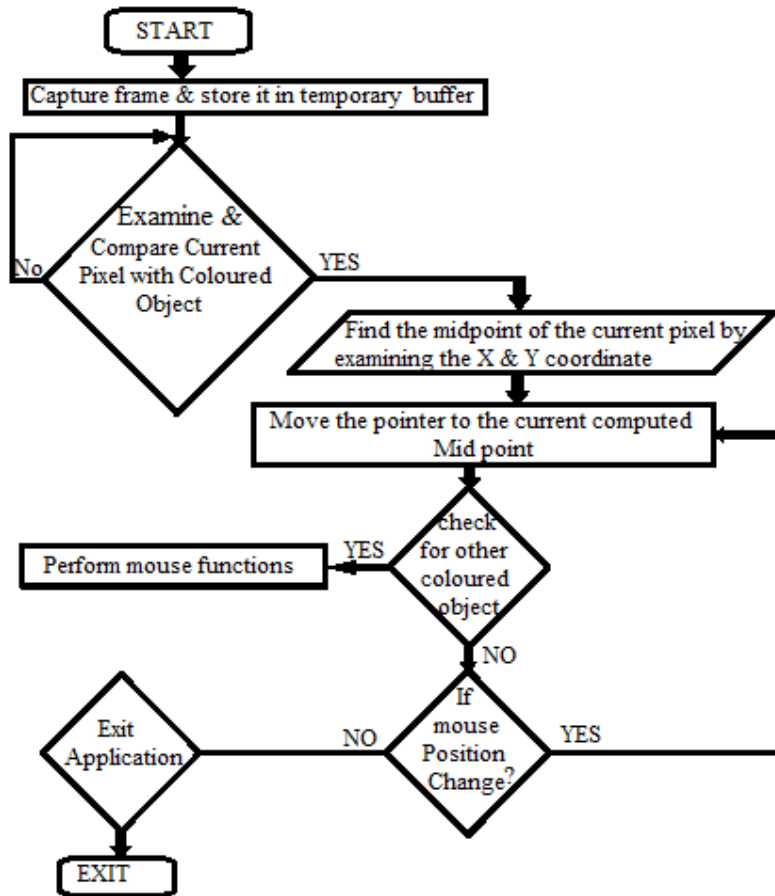


Figure 2. The flow chart

3.3. Functional Requirements

Programmer has to follow a particular sequence of actions to implement their functions. They are Implementing the functions of vlcj to capture the images given by the user ,Store the captured images in a temporary storage buffer, Examining pixel(skipping consecutive four pixels) of the image captured to find the red color, Finding out the mid points by examining x and y coordinates of the current area detected, The pointer reaches the position where the midpoint is found, Examine another pixel containing the red color and find out its midpoint, To move the pointer, compare the current pixel with previous pixel, If the compared two pixels are the same, then the pointer starts moving, Perform mouse functions depending on the length detected between the new computed Pixels.

The User also has certain functions. The functions performed by the user in order are Place a red colored finger cap or any other red colored substance in front of the webcam, Move the substance in front of the webcam to see the pointer moving, selecting icons on the screen and doing mouse functions.



Figure 3. Hand gesture



Figure 4. Hand gesture

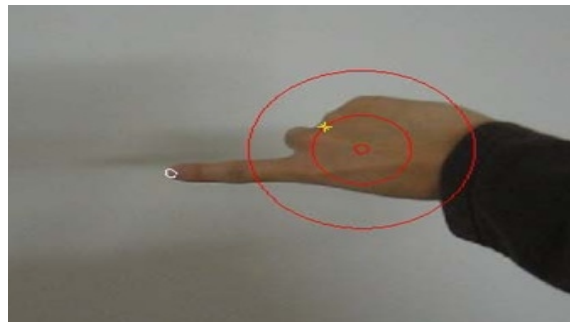


Figure 5. Hand gesture

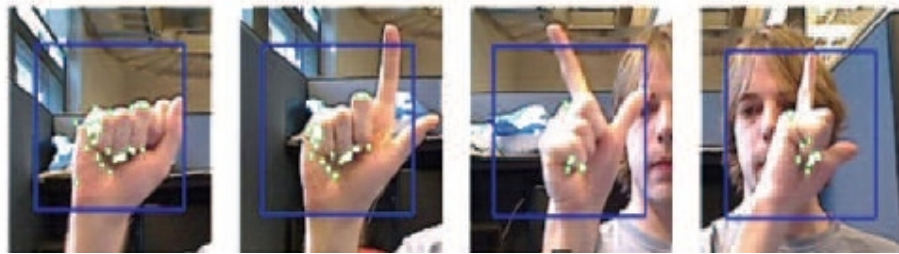


Figure 6. Hand gesture

4. RESULTS

From our implementation and execution of our program we found that the mouse pointer can be made to move and its functions can be implemented without the use of a touchpad or mouse. The pointer is moved with the help of our finger gestures by placing the specific color substance in our hand (any colored cap or any colored small substance) making us easy to use our system works. The performance of the software has been improved.

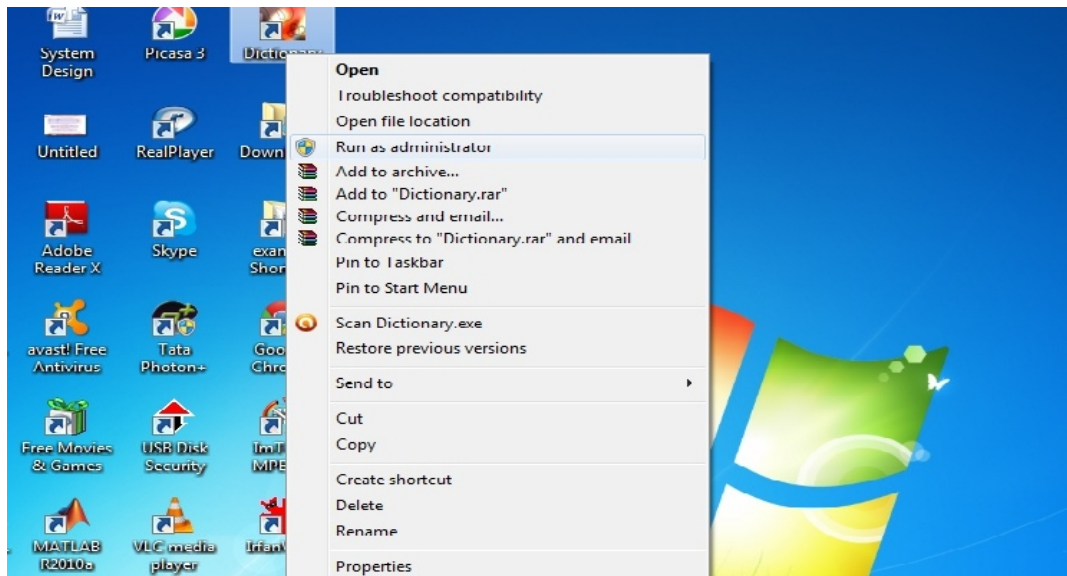


Figure 7. Screenshot for mouse pointer functions in windows xp

The Main problem encountered was when the finger shook a lot. Each time the finger position changed, illumination changed every frame. To fix this problem we added a code to make the cursor not to move when the previous and current red colored pixel difference is within 4 pixels. Technologies we use in this paper assume that all hand movement is properly coordinated.



Figure 8. Screenshot for mouse pointer functions in linux

5. FUTURE WORK

There are still many improvements that can be made to our system like improving the performance of the current system and adding features such as enlarging and shrinking windows, closing window, etc. by using the palm and multiple fingers. The current system is variant to reflection and scale changes and requires proper hand gestures, good illumination technology and powerful camera for the performance of mouse functions. Precision can always be increased at the cost of recall by adding more stages, but each successive stage takes twice as much time to find harder negative samples and the applications which benefit from this technology. We present an image viewing application as an example of where this technology could lead to a more natural user interface. The same could be said for navigating something like Google Maps or browsing folders on a screen. However, the applications reach far beyond that. They are particularly compelling in situations where touch screens are not applicable or less than ideal. For example, with projection systems, there is no screen to touch. Here vision-based technology would provide an ideal replacement for touch screen technology. Similarly in public terminals, constant use results in the spread of dirt and germs. Vision-based systems would remove the need to touch such setups, and would result in improved interaction.

5.1. Enlarge, zoom in and shrink, zoom out

More experiments were done to implement zooming and shrinking. Different hand gestures were analyzed to find the proper gesture for zoom and shrink functions. "Moving hand apart" was the initial hand gesture analyzed but later was discarded due to its limitation of using two hands. The most convenient hand gesture selected was expansion and contraction movement of thumb and index finger. Appropriate scaling factor for enlarging and shrinking are chosen. The scaling

across the thumb and index finger is taken for further procedures. The thumb and index finger movement is done repeatedly to scale the distance to implement these functions.

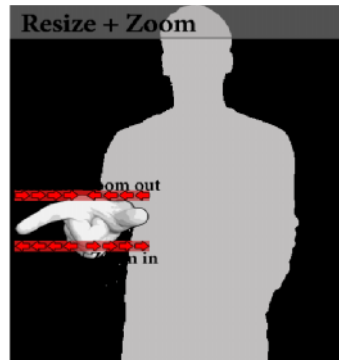


Figure 9. Diagrammatic representation of zoom and shrink

6. CONCLUSION

We developed a system to control the mouse cursor and implement its function using a real-time camera. We implemented mouse movement, selection of the icons and its functions like right, left, double click and scrolling. This system is based on image comparison and motion detection technology to do mouse pointer movements and selection of icon. However, it is difficult to get stable results because of the variety of lighting and detection of the same color in any other location in the background. Most algorithms used have illumination issues. From the results, we can expect that if the algorithms can work in all environments then our system will work more efficiently. This system could be useful in presentations and to reduce work space. In the future, we plan to add more features such as enlarging and shrinking windows, closing window, etc. by using the palm and multiple fingers. The performance of the software can be only improved by small percentage due to the lack of a powerful camera and a separate processor for this application.

REFERENCES

- [1] Hojoon Park. A Method for Controlling Mouse Movement using a Real-Time Camera. www.cs.brown.edu/research/pubs/theses/masters/2010/park.pdf 2010.
- [2] 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.
- [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] Ben askar Chris Jordan, Hyokwon Lee mouse free. <http://www.seas.upenn.edu/cse400/CSE400-2009-2010/nal-report/Jordan-lee.pdf>, 2009-2010.
- [6] Pranavmistry, Sixthsense. <http://www.youtube.com/watch?v=ZfV4R4x2SK0>, 2010. [Online; accessed 28-november-2010].
- [7] Mark dot lee Java bindings for the vlc media player. <http://www.capricasoftware.co.uk/vlcj/index.php>, 2011. [Online; accessed 1-may-2011].
- [8] Pranavmistry, Mouseless <http://www.pranavmistry.com/projects/mouseless/>, 2010 [Online; accessed 28-november-2010].
- [9] Shaker, N.; Abou Zliekha, M.; Damascus Univ., Damascus, Real-time Finger Tracking for Interaction <http://ieeexplore.ieee.org/search/freesrchabstract.jsp> 2007

Authors

Shany Jophin received the BTECH degree from the Department of Computer Science and Engineering, Government Engineering College, Kannur University, Wayand, in 2011. She is currently doing Masters degree in the Department of Computer Science and Engineering, Adi Shankara Institute of Engineering and Technology, MG University, Kerala, India. Her research interests include Cyber forensics, Networking and Information Security.



Sheethal M.S received the BTECH degree from the Department of Computer Science and Engineering KMEA engineering college, Edathala, in 2007. She is currently doing Masters Degree in the Department of Computer Science and Engineering, Adi shankara institute of engineering and technology, MG university, kerala. Her research interests include fuzzy clustering and image processing.



Priya Philip received the BE degree from the Department of Computer Science and Engineering, Sun college of engineering and technology, Nagarcoil , Anna university in 2011. She is currently doing Masters Degree in the Department of Computer Science and Engineering,, Adi Shankara Institute of Engineering and Technology, MG University, Kerala, India



T M Bharguram currently working as assistant professor of Adi Shankara Institute of Engineering and Technology, Department of information technology, Kerala. He has 4 years of teaching experience in various engineering colleges and industry experience in Hiranandani Group, Mumbai, India.

