Project Mudra: Personalization of Computers using Natural Interface

Ankit Dave
Mumbai, India

Yogesh Bhumkar
Mumbai, India

Abin Abraham
Hyderabad, India

Rekha Sugandhi
Associate Professor,
MITCOE, Pune, India

ABSTRACT

In recent years, the popularity for natural interfaces has experienced tremendous growth. Natural interfaces provide the computers or the so called machines with a sense of humanity. The basic knowledge required to understand and utilize a computer system is reduced to a minimal extent. There have been recent commercial developments in the area of natural interfaces, including the Microsoft XBOX Kinect and Nintendo WII. However, the capital cost of such a system remains very high.

The primary intention of this paper is to design a technique to redesign every aspect of our interaction with the computer. It explores and develops an interface that helps simple computer systems to support gestural interface. We could utilize the low cost cameras available with the laptop or desktop to support detection of human gestures and multiple touch experiences, which would ultimately lead to the direct interaction of computers with humans. In such a system, the need for external input devices like mouse and keyboard could be eliminated. The paper discusses how such a system can be implemented, along with a brief description of the approach used.

General Terms

Human Computer Interaction, Image Processing

Keywords

Mouse Movement, Computer Vision, Mouse less, Keyboard less, Swipe Gesture, Air Swipe, Human Computer Interface, Usability engineering, User Interfaces, Ergonomics, Natural Interface

1. INTRODUCTION

Interaction of users with the computers has been changing ever since the computers were made. It all started with a command line interface, which made computers difficult to use. Hence only people with prior knowledge of computer commands can use it. With the invention of mouse and Graphical User Interface (GUI), interaction with the computer became so simple that it made it easily accessible to a common user. Mouse and GUI made computers personal, in real sense of the term.

Even though Mouse and Keyboard were easy to use, the interaction which they offered was far from being natural. Moreover, when user has to interact with the computer he has to come near the computer. There were two parts in the life of a user, one where it is doing its day-to-day activity and other is where he is using the computer. The way it interacts in both these worlds was totally different.

There have been recent commercial developments in the area of natural interfaces, including the Microsoft XBOX Kinect [1] and Nintendo WII [2]. However, the capital cost of such a system remains very high.

This paper suggests how interaction between computer and user can be made more natural. In such a system the need for external input devices like mouse and keyboard will be eliminated, keeping the overall cost of the system very low. This will make it accessible to a common man.

1.1.Approach to the problem:

The above stated system is implemented in two approaches:

1.1.1. Using Gloves with Color markers:

In this approach, mouse movements will be detected using color caps. User will have to wear a glove with color markers. Specific color on the screen will be tracked and converted to mouse movement.

1.1.2. Using a Depth Sensor:

In this approach, we removed the dependency on the color marker. Thus with hand movements user can control the mouse. Moreover, the use of depth sensor makes it possible to perform gestures in 3D.

1.2.Features of the system:

1.2.1. Mouse Movement [3]:

Mouse is an integral part of the current Graphical User Interfaces, which is employed in majority of the operating systems. The project enables us to control the movement of cursor on the screen, using the motion of our fingers.

1.2.2. Single Click (primitive pattern) [3]:

In the case of coloured caps approach, the single click is triggered by moving the thumb close to the index finger. The caps are identified by coloured caps. In depth sensor, single-click is triggered by pushing the palm in the air towards the sensor.

1.2.3. Double Click (multiple primitive patterns) [3]:

userFigure 1: Approach 1 with color markers

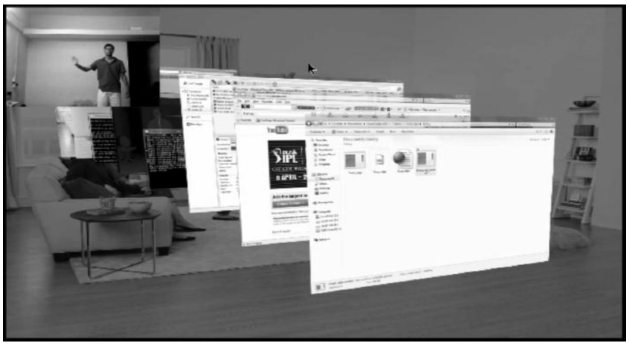

Figure 2: Approach 2 with depth sensor

The double click operation is triggered by two simultaneous single clicks.

1.2.4. Right Click (primitive pattern) [3]:

The right click is simulated by the showing the middle finger's cap along with index finger. The duration of the visibility of the finger determines the duration of right click operation. In case of depth sensor, right click is performed by raising the other hand up, and waving.

1.2.5. Drag-n-Drop (multiple primitive patterns) [3]:

The drag-n-drop is simulated by the clicking (keeping the thumb closer to the index finger), dragging the cursor and then releasing the click (moving the thumb away). In case of depth sensor, it is performed by clicking on an object then moving your hand in air to drag, and then moving your hand away from the sensor in Z-axis.

1.2.6. Air Swipe Operation [4]:

The swipe operation is the method of simulating certain actions according to the cursor movement at a velocity which greater than the threshold value. For example, the up-swipe consists of moving the cursor bottom-up at a high speed, without any triggering gesture.

1.2.7. Air Keyboard [5]:

This is a virtual keyboard, which can be used as a substitute to a normal hardware keyboard. This is specially designed for systems with gestural interface. It has capability of providing suggestions and it makes typing faster.

1.2.8. Line Gestures:

The line gestures are fast and easy to learn. They are recognized by the application trivially. We used the concept of slope of lines in implementing the line gestures. Examples would include making gestures of A, T, 4, etc.

2. RELATED WORK

Hojoon Park [7], used hand to perform mouse operations without wearing any color caps. User moves mouse pointer by moving the index finger. To perform left click, user keeps angle between thumb and the index finger between 70 and 90. The right click event occurs when left click event is kept for more than 3 seconds. As talked in discussion of [7] they faced problem hand and finger tips detection. Detection of hand and fingers involve algorithms which are complex which leads to reduction in frame rate. We used fixed color markers which speeds up the process of detection of pointing gesture. And in case of depth sensor, we focused only in the region where the hand was last seen, making detection much faster.

Chu-Feng Lien [8] trained their system for recognizing palm and then used palm for mouse movement. For clicking user has to stay at the same position for specific time which then triggers a mouse click. It is done by checking density of palm for a point over some consecutive frames. This also has problem with detecting palm correctly. User has to avoid moving his hand continuously so that unwanted click event is not triggered.

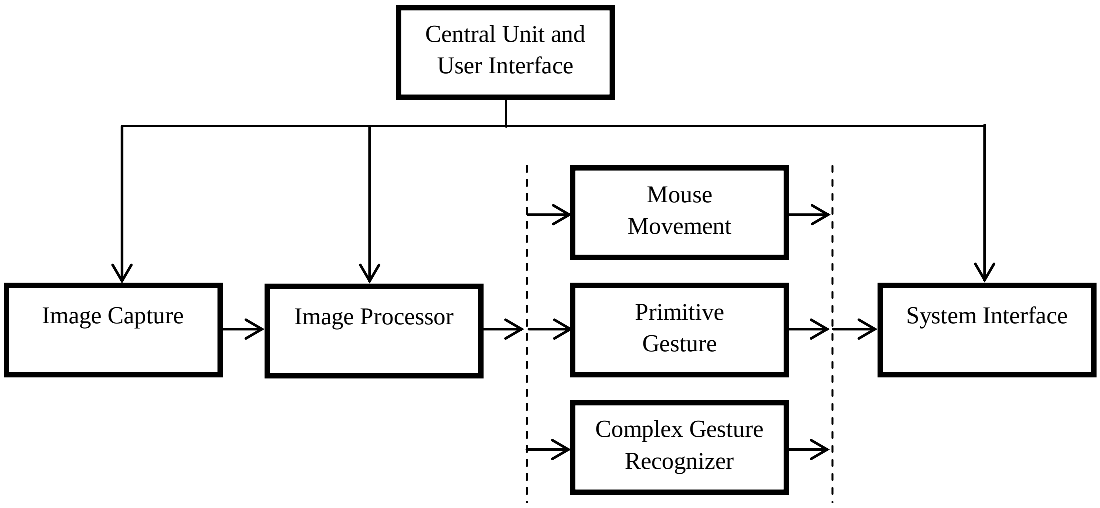

Figure 3: Flowchart for the system

Paul [9] developed gesture based system for using their Kiosk interface. They trained their system to detect different hand signs out of which they used "thumbs up" sign for mouse movement and "fist" for clicking. These signs can be used for using Kiosk interface. It provides only basic operations viz. mouse movement and left clicks. For other operations user will have to use mouse or keyboard.

In Sixth Sense Technology [10], Pranav Mistry talks about new wearable device which can be interacted with hand gestures. It has a camera for gesture detection and a projector to display on any surface. It supports gestures for clicking, photo capture, and zoom in and out. This system is though very useful, it costs around 350$ [11] and user can perform these operations on application supported by WUW.

Robust hand gesture recognition with kinect sensor[12] system is able to detect palm very clearly which kinect sensor is not able to do. In our system, we are able to perform the same operations without getting into too much details of the hand. This helps us in making the system faster by reducing the processing required per frame. Which ultimately leads to higher frame rate.

3. IMPLEMENTATION APPROACH

Figure 3 shows the system architectural block diagram of project mudra. The software architecture consists of:

3.1.Central (user interface) Unit:

The central unit controls the settings for the project. It provides a unique interface to change the settings of the mouse pointer, add, delete and modify various gestures, train gestures, add, delete and modify various modes of operation and map gestures to modes. The central unit saves all data to a database file, which is then used by the sub-processes for image processing.

3.2.Image Capture:

The image capture unit captures the bitmap image from the web-camera. The output of the camera could be a bitmap of any resolution. For convenience, we have mapped it to 640x480 pixels (which could be changed anytime).

3.3.Image Processor:

The normalization of the bitmap output of previous module occurs here. The image processor extracts the detected points and outputs all of the points in the form of integers. The normalized for of two points at (11, 23) and (56, 78) is 0 11 23 1 56 78.

3.4.Parallel stages of operation:

3.4.1. Mouse Movement module:

The mouse movement module is activated almost all the time. It tracks the mouse pointer over the screen. It is deactivated only when complex gestures are detected.

3.4.2. Primitive Pattern Recognizer module:

The primitive patters are the single clicks and right click gestures. This module is run in parallel but at a rate less frequent than the mouse movement module.

3.4.3. Complex Gesture Recognizer module:

The complex gestures are recorded only when we activate them. This module is not active all the time since it is very time consuming. The actual pattern recognition starts when we tell this module to stop recording and then recognize. In this way, again less CPU resources are utilized.

3.5.System Interface:

The system interface is the actual module which passes on the commands of the previous modules to the operating system. It interacts with the operating system. As a result, this module is treated as an interface and must be implemented for different operating systems, like the Windows, Linux, Mac, etc. These parts along with the user interface are the only OS dependent parts in the architecture.

In the above system, Central Unit constantly processes images, captured by the camera. It extracts the information from the image, to find the location of the hand or finger. This location of the hand or finger is then used to continuously move the mouse cursor, while at the same it tries to find any primitive or complex gesture by the user. Air swipe [4] is one such example of gestures that are detected by the system. After detection of the gesture the system performs the action mapped to them, using System Interface unit.

4. RESULT

In our first approach, we used color capped gloves. Even though the detection and filtering of colors were pretty challenging, we implemented basic image processing techniques background elimination to nullify the colors, corresponding to the color caps, in the background. We performed operations like the mouse movement, mouse click, mouse double click, mouse right click, swipe operations.

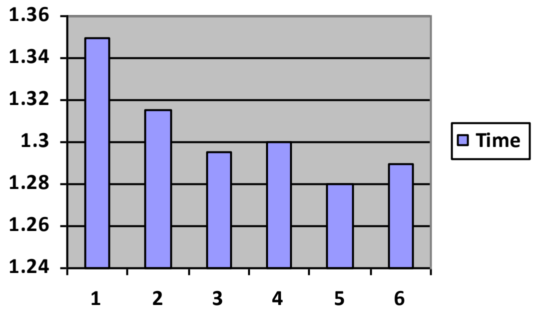

Figure 4: Chart for time required for a sample

In our second approach, we integrated the Kinect sensor (as a depth sensor) and used the RGB camera and IR depth sensors to create a more productive, responsive and advanced version of the system. We also incorporated 3D gestures like the mouse single click (by pushing a button in the air).

We carried out various experiments to determine the performance. We wanted the system to be sensitive enough to perform all the mouse operations and gesture operations with minimum ease. We also wanted the system to have a high response time.

We conducted various experiments separately. In the first experiment, one user was selected, and was asked to perform a mouse left click operation on an icon, starting from one corner screen and the icon is at the diagonally opposite corner. In this experiment, time (in sec) required to perform the operation was calculated. This process is repeated various times to get an average value. We then asked the user to do the same operation with a mouse. And then compared these two average values. To our surprise, they were very close to each other.

In the other experiment, people from different age groups, were selected. They were asked to perform the same experiment.

Figure 5: Chart for time required for various age groups

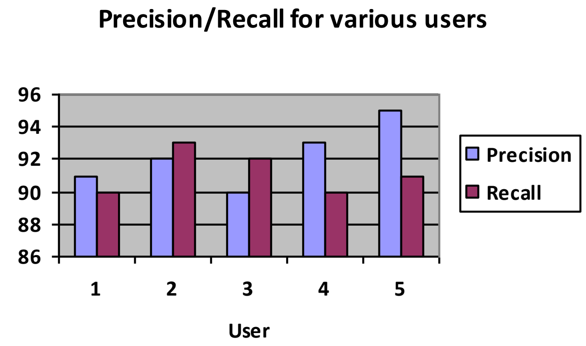

Figure 6: Chart for Precision/Recall

This helped us in understanding how the performance of the system changes with the age of the user. From the graph it is clear, that as the age of the user increase, performance of the system decreases.

Using the performance values in all the features of the system were tested for accuracy. We have used two metrics for calculating the accuracy of a feature. They are precision and recall.

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

tp – True Positive
fp – False Positive
fn – False Negative

These two metric values were calculated for various users, and then we took average of all the values to give a final answer. From figure, it is clear that we were successful in getting high precision of 92.18% and recall of 91.3%.

We then measured the speed of the system in terms of the number of frames it can process per second. The camera used in the system has a frame rate of 30fps. We measured the time taken to process a single frame, with the help of a small program which feeds an image to the system and calculates the time taken to get back the result. The measured value of average time taken to process a single frame is 2.236 seconds. Since this time is small, our program can process at roughly 27fps. The hardware used is Intel Core2 Duo processor, and 2gb RAM, which is a normal configuration in today's world.

For gesture recognition, we measured the precision of the system in detecting the gesture, as compared to the number of input images used to train the system. As shown in figure 7, we observed that as the number of training images was increased, the precision of the system got better, and after a certain point it became almost constant. We were able to achieve average precision rate of 80% which means in 8 out of 10 cases, right gesture was detected.

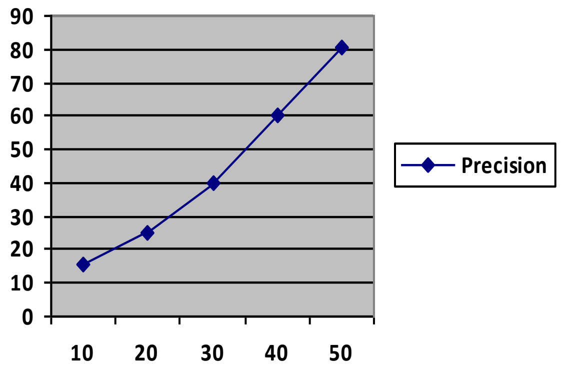

Figure 7: Chart for Precision with increasing training data size

5. FUTURE SCOPE

This system can be used in making the next generation personal computers that do not require any external controller. Present User Interfaces are made for pointing devices. In future, we might have to reimagine them for such systems. We can have a computer which can be controlled completely with our gestures without the need of any external controlling device. We can take these low cost systems to rural places. Children could seamlessly interact with computer system in a very natural way. It can also be used for speech challenged people with sign language support. Presently this system is supported only on windows operating system. In future, we can take it to multiple platforms.

6. CONCLUSION

With the invention of mouse and Graphical User Interface (GUI), our way of interaction with the computers got totally changed. Since then, for so many years, mouse is the only means of interaction. But, the way of interaction mouse provides is not natural. We don't interact with any other thing in our life like we do with our computer.

The work discussed in this paper, is an attempt to redesign every aspect of our interaction with the computers. We have covered everything from mouse, to keyboard, to gestures. Result shows that we are able to achieve a precision of 92.18% and a recall of 91.3%. Thus, this system can totally change the future of personal computers.

7. REFERENCES

[1] Kinect-Xbox n.d. Web 2011 < http://www.xbox.com/en-US/kinect>

[2] T. Schlomer et al., "Gesture Recognition with a Wii Controller", in Proceedings of the 2nd international conference on Tangible and embedded I.

[3] YouTube - Broadcast Yourself n.d. Web 2011 <http://www.youtube.com/watch?v=UJox-zQgOBk>

[4] Rekha Sugandhi, Abin A Alichan, Ankit Dave and Yogesh Bhumkar. Article: Air Swipe: A Modified Swipe Gesture System.International Journal of Computer Applications 26(7):9-12, July 2011. Published by Foundation of Computer Science, New York, USA

[5] YouTube - Broadcast Yourself n.d. Web 2011<http://www.youtube.com/watch?v=SMckgO3Hzf>

[6] YouTube - Broadcast Yourself n.d. Web 2011< http://www.youtube.com/watch?v=zVzgpjJM1Ds>

[7] Hojoon Park. A Method For Controlling The Mouse Movement using a Real Time Camera, Brown University ,Providence ,RI ,USA, Department of computer science, 2008

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

[9] Paul Robertson, Robert Laddaga and Max Van Kleek. 2004. Virtual mouse vision based interface. In Proceedings of the 9th international conference on intelligent user interfaces (IUI '04). ACM, New York, NY, USA, 177-183. DOI=10.1145/964442.964476 http://doi.acm.org/10.1145/964442.964476

[10] P. Mistry, P. Maes. SixthSense. A Wearable Gestural Interface. In the Proceedings of SIGGRAPH Asia 2009, Sketch. Yokohama, Japan. 2009

[11] YouTube - Broadcast Yourself n.d. web 2011< http://www.pranavmistry.com/projects/sixthsense/>

[12] Zhou Ren , Jingjing Meng , Junsong Yuan , Zhengyou Zhang, Robust hand gesture recognition with kinect sensor, Proceedings of the 19th ACM international conference on Multimedia, November 28-December 01, 2011, Scottsdale, Arizona, USA [doi>10.1145/2072298.2072443]