

Air Swipe: A Modified Swipe Gesture System

Rekha Sugandhi
Associate Professor,
MITCOE
Pune, India

Abin A. Alichan
MITCOE
Pune, India

Ankit Dave
MITCOE
Pune, India

Yogesh Bhumkar
MITCOE
Pune, India

ABSTRACT

Traditional systems with pointing devices, lack swipe gestural interaction. Nowadays, there are a lot of new pointing devices which provides the facility for swipe gestures. But they lack user friendliness. In these systems, the user has to interact with the system in a way which is different from traditional methods. A very good example is Apple's Mac Book touchpad, in which the user has to use three fingers to perform a swipe; for doing mouse operations only one finger is required.

This paper focuses on the implementation of the air swipe feature for the above mentioned systems, which does not require any external adaptation on the user's part. The paper discusses how the swipe gestures could be performed without interfering with the normal pointing-device operations, along with a brief description of the approach used to implement this feature.

General Terms

Human Computer Interaction

Keywords

Swipe Gesture, Air Swipe, Human Computer Interface, Usability engineering, User Interfaces, Ergonomics

1. INTRODUCTION

Swipe technology was introduced by Apple INC, in its product called iPhone. It is a multi touch mobile phone, which supports swipe gesture [1]. With this feature, user can browse through images with a swipe on the screen. There could be four different kinds of swipe motions that are possible, depending on the direction in which they are performed; these are, left, right, up and down [1].

Later on, this technology was extended to the laptops and the desktops of Apple. But these systems were already having a pointing device and introduction of the swipe feature raised a problem. How would the user perform a swipe on a pointing device? Performing a swipe on a pointing device will move the cursor in that direction. So, how would the computer distinguish between a normal mouse movement and a swipe? To solve this problem, apple came out with a solution. In their pointing device, with just a single finger, the user could perform normal mouse operations. While, on the other hand, for doing a swipe gesture, the user would have to use three fingers. Thus with the use of three fingers, the computer comes to know that the user has to perform a swipe [2]. This method solves the problem for the computer, but it restricts the usage of this technology to devices which supports multi-touch. What about the ones with a single touch?

In Apple's interface [3], the user has to switch between one and three fingers each time to switch between pointing operation and swipe. If this switching is frequent then this can introduce errors from user by using swipe [4].

This paper suggests solutions to above problems with the help of swipe technology as –

- 1) Same method of interaction for pointing device operations and swipe feature, removing the need of switching between modes.
- 2) Can be used on single touch pointing devices.

In this current implementation of air swipe, user uses pointing devices to point mouse pointer and perform clicking, which is the traditional operation. At any point of time, in order to perform swipe operation the user simply needs to move the pointer in the required direction rapidly, with more speed than normal pointer movement. This speed is predetermined and can be tuned according to the user. Thus, with a normal speed user can perform pointing device operations and with a rapid movement of his fingers, a swipe can be performed.

2. RELATED WORK

Opera is a web browser provided by Opera Software ASA. This browser is available for various platforms including Windows, Linux, Mac and mobile devices. It has built in feature of swipe gesture called as opera mouse gesture [5]. In opera the user can use browser in traditional way as with other browser. If he wants to perform swipe he presses right button and moves mouse pointer while holding the right click. For each swipe operation that user performs, it starts with clicking right button. Opera has provided various predefined swipe gestures such as left for previous page, right swipe for forward, down for open new tab and many more [5].

It is a good way to introduce the swipe feature in such systems where it is required. But it restricts the use of swipe in browser, as user can only use it in browser.

Another example is Mouse Gesture. Mouse Gestures is an open source Java library for recognition and processing mouse gestures provided by Smardec [6]. To perform a gesture, the user can hold down a mouse button and move the mouse in a predefined direction. The mouse button which is to be held can be left, middle or right button. Developer who wants to use these gestures in their application can use this java library while developing the application [6].

It gives user very good way to interact with the application but it limits the use of gesture to that application. As developers can

develop their own gestures it can lead to inconsistency among gestures between applications.

Bezel Swipe is a swipe gesture as in [7]. This gesture is implemented for touch screen devices. Bezel swipe is used to provide features like multiple selection, cut, copy, paste without interfering with zoom, pan, tap which are usually performed. To perform the bezel swipe user moves finger from leftmost screen area called bezel to the target object. After completion of this gesture the object gets selected. Same operation on another object will select that object, thus availing multiple selections. Similarly, to select a text this operation has to be performed twice; once to mark start of text and then to mark end of text. Normal swipe operation on text scrolls text while bezel swipe selects it.

Contactless gesture recognition for mobile devices[8] introduced gesture recognition such as swipe using the proximity sensors built in mobile. The user moves finger over the sensor and gesture is detected. This gesture can be separated from the touch gesture and other operations which are not possible by touch swipe can be performed from contactless swipe gesture. Thus by this way new swipe gesture is added without affecting original swipe concept.

3. IMPLEMENTATION APPROACH

The pointer and swipe operation can be distinguished from each other, depending on the speed with which the user is moving it. Pointer is continuously tracked, and its position is recorded, which is then used to calculate its instantaneous velocity. This instantaneous velocity can be plotted with reference to time. Figure 1 shows graph of instantaneous velocity verses time.

There are two regions in which the user can operate. First is the normal pointer operation region, this is the range of instantaneous velocity which is allowed for pointer operation. And the second region is where the instantaneous velocity crosses the threshold. In this region, the swipe gesture is performed, depending on the direction. Thus we divide our instantaneous velocity bandwidth into two parts for two different operations.

There are two types of thresholds defined for instantaneous velocity.

1) Triggering threshold limits

It consists of two sub threshold limits, namely upper and lower. When velocity of pointer crosses lower threshold limit it activates the recording of direction of pointer. To trigger a swipe the velocity must fall between upper and lower limits of triggering thresholds, to ensure that the velocity is neither too less for the swipe nor too fast for the system. As shown in figure 1 at point p1 the velocity is crossing lower threshold limit activating recording of gesture.

2) Cutoff threshold

After the swipe is triggered and recording of direction is started velocity is tracked to check whether it falls below cutoff threshold. If it falls below cutoff threshold then one swipe action is performed depending on the recorded direction. As shown in figure 1 at point p2 velocity is falling below cutoff threshold.

Threshold values are determined by performing various experiments. Firstly, the user is allowed to perform a swipe, and the velocity value is recorded. Using these values the various threshold values can be determined. These experiments are performed on various users, so that the threshold values are more accurate.

Using machine learning, the values of thresholds can be adjusted according to user [9]. If user is an older person, who may not be comfortable to move the pointer fast enough to perform swipe. In that case, the values of lower and upper triggering threshold can be lowered to make it easy for the user. In exactly opposite case, if user is very active and he moves the pointer rapidly, then the triggering threshold values can be increased to avoid false swipe detection. Thus with machine learning, the system continuously changes itself according to the user, by modifying the values of the threshold.

From a user's perspective, to do a mouse operation, the user can move the pointer with a normal speed. For swipe, the user just needs to move the pointer in that direction rapidly. This makes the job simple, as the user doesn't have to change the way of interaction for performing different operations.

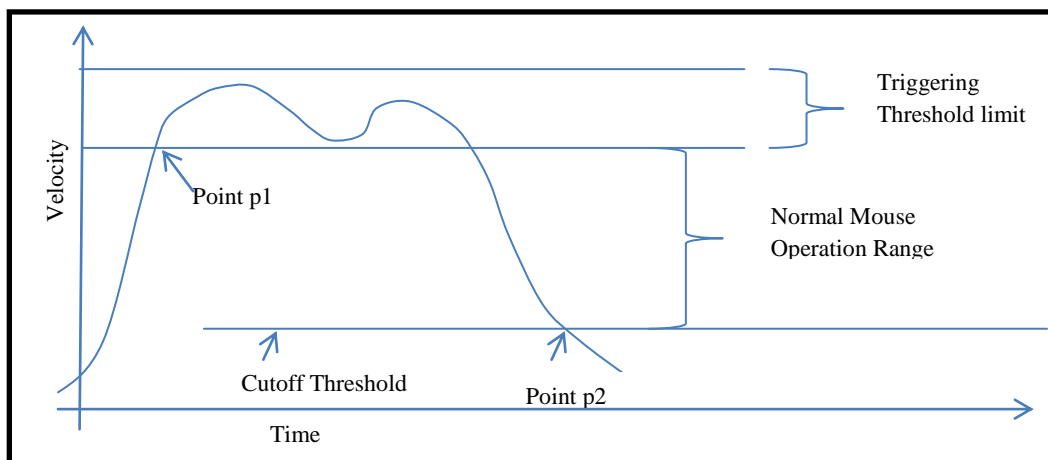


Figure 1: Graph of velocity versus time

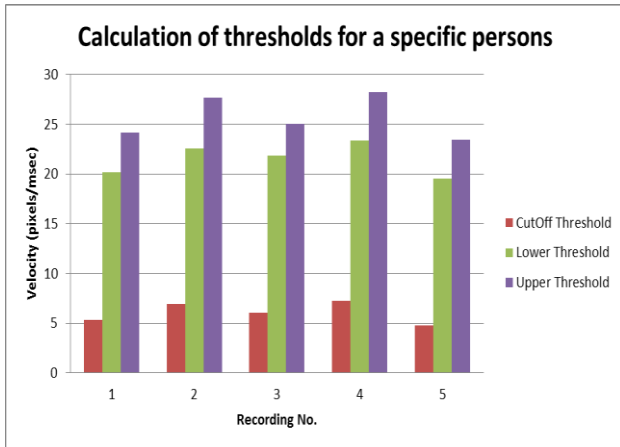


Figure 2: Chart of velocity for three thresholds

4. RESULT

Several experiments have been carried out in order to determine the values of the threshold. These experiments were conducted in various stages. In the first stage, one user was selected, and asked to perform a swipe operation. In this process, the threshold values have been measured. The same experiment is repeated, various times, to get an average value. Thus the final values, is the average of all the values.

In the second stage, people from different age groups were selected, and then measured threshold values for each age group. This helped us in gaining an understanding of how the threshold values changes with respect to a person’s age. From the graph, it can be observed that, as the age increases the threshold values decreases.

Using the above gained threshold values the swipe feature was tested for accuracy. Two metrics have been used for calculating the accuracy, namely precision and recall for various users [10].

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

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

Where tp- True Positive

fp- False Positive

fn- False Negative

The precision and recall values for various users have been calculated and then the average of all those values and been taken to give the final answer.

Thus, from the figure 4, it can be observed that high precision and recall has been achieved.

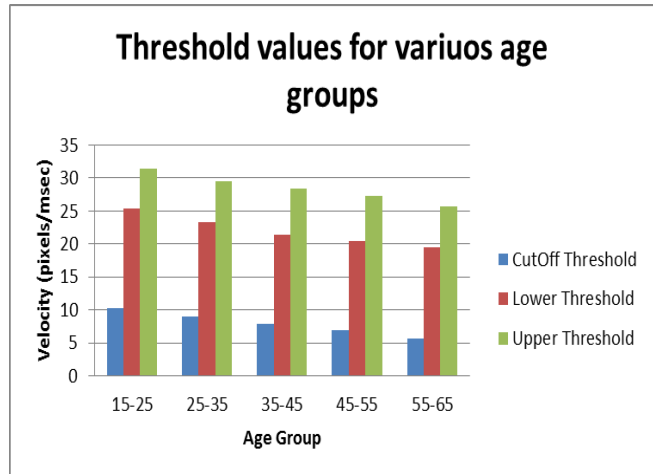


Figure 3: Chart of velocity for three thresholds for various age groups

5. FUTURE SCOPE

This swipe method of implementation can be used in various futuristic products, like a system in which mouse is being operated with hand movements using a camera [11,12]. Thus hand movements at a normal speed will correspond to mouse movement, while rapid movements will corresponds to swipe. In these systems, with the help of depth sensor, more information about the user can be obtained. This information can be exploited to provide various features. This method of splitting the bandwidth of the instantaneous velocity in various parts, for various operations, can be helpful in providing a

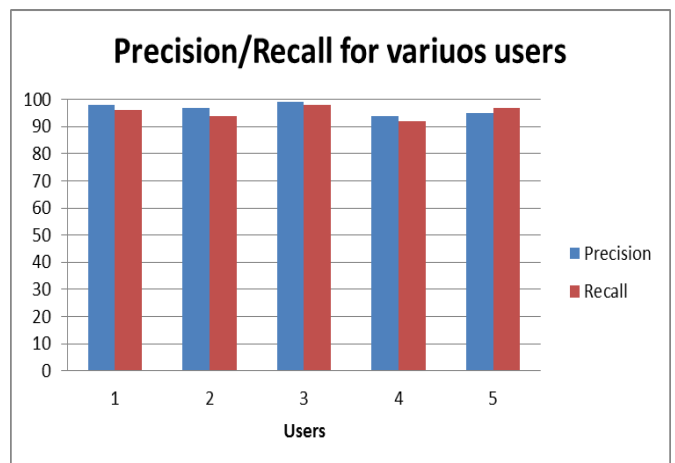


Figure 4: Chart of Precision/Recall calculated

uniform interaction between a human and a computer.

6. CONCLUSION

With the introduction of swipe feature by Apple in iPhones the way human interact with touch enabled devices has changed. A lot of manufacturers adopted this method in their subsequent products. But, with the use of this feature in single touch pointing devices, raised a problem. User has to interact with

computer in order to perform a swipe, in a different way, than his normal operations.

The work discussed in this paper attempts solve this problem of user interaction, by making a system in which there is no difference, in the way user interact, to perform all his previous operations along with swipe. Result obtained from performing several experiment, shows that 93.2% precision and 91.6% recall, can be obtained. Thus, it provides a new and easy way of interaction of human with the computer by removing the extra overhead on user side.

7. REFERENCES

- [1] Sadun, Erica. "Taking Your iPhone to the Max". Apress, 2008.
- [2] Noelle, Robin. "How to Do Everything MacBook". McGraw-Hill, 2010.
- [3] "Apple - Magic Trackpad", Apple Inc., Web 2011. <http://images.apple.com/magicktrackpad/>
- [4] Raskin, Jef. "The Humane Interface: New Directions for Designing Interactive Systems". Addison-Wesley, 2000. 50.
- [5]"Mouse Gestures in Opera." *Opera Tutorial*. Opera Software ASA, 2011. Web. 2011. <<http://www.opera.com/browser/tutorials/gestures/>>.
- [6]"Mouse Gestures." *Smardec-Products*. Smardec, n.d. Web. 2011. <<http://www.smardec.com/products/mouse-gestures.html>>.
- [7] Volker Roth, Thea Turner, "Bezel Swipe: Conflict-Free Scrolling and Multiple Selection on Mobile Touch Screen Devices", In Proceedings of the 27th international conference on Human factors in computing systems (CHI '09). ACM, New York, NY, USA, 1523-1526. DOI=10.1145/1518701.1518933 <http://doi.acm.org/10.1145/1518701.1518933>
- [8] Heng-Tze Cheng; An Mei Chen; Razdan, A.; Buller, E.; , "Contactless gesture recognition system using proximity sensors," *Consumer Electronics (ICCE), 2011 IEEE International Conference on* , vol., no., pp.149-150, 9-12 Jan. 2011 doi:10.1109/ICCE.2011.5722510 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5722510&isnumber=5722481>
- [9] Chen, Hsinchun. "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms". Journal of the American Society for Information Science. 2004.
- [10] Olson, David, and Dursun Delen. "Advanced Data Mining Techniques". Springer, 2008. 138.
- [11] *Project Mudra*. N.p., 2011. Web. 2011. <http://projectmudra.blogspot.com/2011_03_01_archive.html>.
- [12] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. "Vision-based hand-gesture applications". *Commun. ACM* .February 2011