TELKOMNIKA, Vol. 11, No. 5, May 2013, pp. 2710 ~ 2715
ISSN: 2302-4046

Real Time Break Point Detection Technique (RBPD) in Computer Mouse Trajectory

Seyed Yashar Banihashem[1], Nor Azan Mat Zin*[1], Noor Faezah Mohd Yatim[1], Norlinah Mohamed Ibrahim[2]
[1]School of Information Technology, Faculty of Information Science & Technology,
Universiti Kebangsaan Malaysia 43600 UKM Bangi, Selangor D.E, Malaysia
Ph: 603 89216814, Fax: 603 89256732
[2]Department of Medicine, Universiti Kebangsaan Malaysia Medical Center, Jalan Yaacob Latif, Bandar Tun Razak, Cheras 56000 Kuala Lumpur.
*Corresponding author, e-mail: azan@ftsm.ukm.my

Abstract

This paper presents a new technique for detecting break points in computer mouse trajectories called Real time Break Point Detection (RBPD). The mouse trajectory from motor-disabled people contains a lot of break points. RBPD can detect these break points by adapting to the user's tremor level. To test the effectiveness of this new technique, an experimental study involving five participants suffering from Parkinson's disease (PD) was conducted. The test results showed that RBPD technique is effectiveness in detecting break points.

***Keywords**: Break points; Parkinson disease; HCI; Mouse Trajectory*

1. Introduction

The pathological tremor is known to be the most common movement disorder. It can also be defined as an involuntary movement with approximately rhythmic and roughly Sinusoidal repositioning [1].

One of the common input device for computer interaction is the computer mouse [2-5]. Involuntary tremors affect mouse movements and for people with Parkinson's Disease (PD)s, the computer interaction tends to be difficult. This is evident when their tremors induce movement that changes the mouse direction from its intended path. In order to correct this error, attempts are made to re-position the cursor to its initial location. The corrections and changes made are referred to as an unsmooth mouse trajectory.

In other words the mouse trajectory is dotted with break points (Figure 1). By detecting these points, it is possible to learn about the users' mental state [6], stress level or even their health condition [7]. These points can also be used to feed noise-filtering techniques; whereby the break points are removed and a smoothed mouse trajectory is delivered to the users. This paper describes the real time algorithm for accurate detection of break points in the user's mouse trajectory.

The detected points can then be used at any stage of the user behavior investigation The algorithm is named "Real time Break Point Detection" (RBPD). The technique is evaluated by users with PD, for its functionality.

2. Research Method

In two main steps we analyzed the effect of tremor on mouse trajectory. By analyzing users' mouse movements, an obvious visual difference between a healthy and a disabled person's mouse movements were observed; disabled people had more break points in their mouse movements compared to normal users. These break points are caused by unwanted tremors, as illustrated in Figure 1.

Received January 20, 2013; Revised March 18, 2013; Accepted March 25, 2013

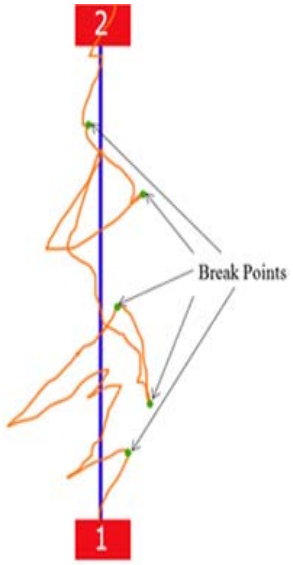

Figure 1. Break points in the trajectory

To detect break points in the trajectory, an algorithm was designed. From another point of view, each break point can also be defined as a turning point in cursor movements. At any point of time, the mouse cursor can move in nine different coordinates. Figure 2 shows these possible nine conditions.

Figure 2. Possible positions of mouse movement

3. RBPD Algorithm

Mouse movements were analyzed to detect break points. A break point must satisfy two conditions. First, the cursor-moving vector changes in the last 0.5 seconds. For example if it was moving to the right, it suddenly moves up or down. To satisfy the next condition, the cursor should have enough position changes during the last 0.5 seconds. All these steps are shown in Figure 3.

The position of the mouse cursor on a computer screen is in X and Y format. X refers to cursor's horizontal position that starts from zero on the left side of the screen. Y refers to cursor's vertical position from zero to the top of the screen. In Windows-based computers, the top left point of the monitor is the start point for X and Y (X=0, Y=0) and the bottom right point is the end point for X and Y (X=Max, Y= Max). Max depends on the monitor resolution capacity. To detect direction changes during the mouse movements a sub-method, which is a part of break point detection method was designed. This sub-method will be explained in the next section.

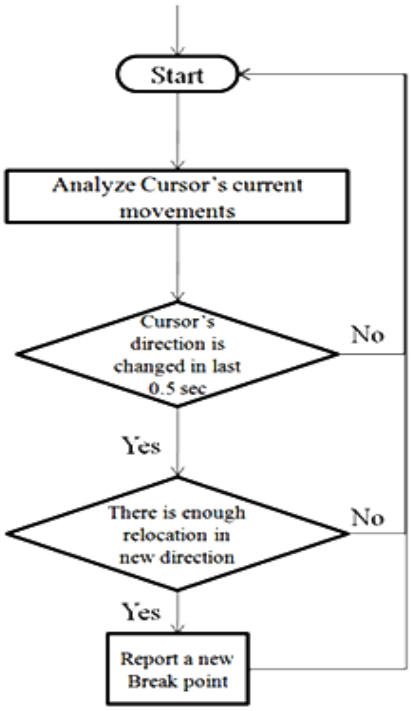

Figure 3. Break point detection algorithm

3.1. Cursor Direction Rules

To detect direction changes, the cursor coordinates (X, Y) were sampled and analyzed continuously. As explained, the cursor's coordinates are accessible in X and Y format. By comparing two sampled points, the current cursor direction could be obtained. Based on X and Y changes, the cursor has only nine different directions in which it can move. All these possible situations are explained in Table 1.

Table 1. Comparing points

	X	Y
Stop	Not Changed	Not Changed
Right	Increased	Not Changed
Left	decreased	Not Changed
Up	Not Changed	Decreased
Down	Not Changed	Increased
Up Left	Decreased	Decreased
Down Left	Decreased	Increased
Up Right	Increased	Decreased
Down Right	Increased	Increased

The table was created based on the computer operating system (Windows) coordinates. Despite this format, the standard coordinate diagram yields negative values but the computer screen showed positive values only and the point (0,0) is not at the center of the diagram. After comparing the coordinates of two sampled points, if the first sampled point has a different value of X or Y compared to the second sampled point, the cursor's moving direction will be detected as per Table 1. In this step, the method will compare the cursor's current direction to its previous one to find out changes in cursor direction. These steps are displayed in Figure 5 using one mouse path sample and its related code.

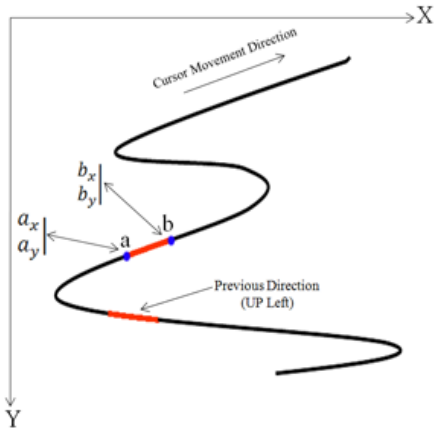

Figure 5. Sample mouse trajectory

3.2. Role of Tremor Level in RBPD

The codes sample shown in Figure 6 is used to detect turning conditions. After this step, the cursor must be checked to see if it had enough physical movement during the past half-second. Based on the primary sampling, there was a difference in the number of break points for low and high tremors.

```
if (bx > ax)
        {
          if (by == ay) { Direction = "Right"; }
          else if (by < ay) { Direction = " Up Right"; }
          else { Direction = "Down Right"; }
        }

        if (bx == ax)
        {
          if (by < ay) { Direction = "UP"; }
          else if (by > ay)
          { Direction = "Down"; }
          else { Direction = "STOP"; }
        }
        if (bx < ax)
        {
          if (by < ay) { Direction = "Up Left"; }
          else if (by == ay) { Direction = "Left"; }
          else { Direction = "Down Left"; }
        }
```

Figure 6. Codes sample to detect turning

In low-level tremors, there were more physical repositions after and before the break point. This means that, many break points could potentially be neglected in high tremors if the value of repositioning is not adjusted to tremor levels.

Figure 7 further explains this issue in visual form. Respectively Fig 7 (a) represent a trajectory from user with low level tremor and Fig 7 (b) represent a user with high level of tremor.

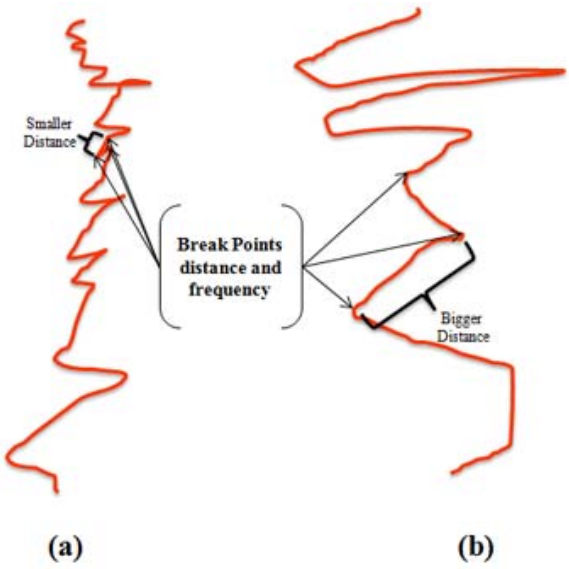

Figure 7. Low (a) and High level (b) tremor difference on mouse trajectory

Coordinate values on the computer screen yielded positive values as a mathematic absolute function to get exact movement was used during the subtraction of sampled points. Formula 1 is used to calculate the repositioning amount and makes decisions about reporting the break point.

$$|b\_x-a\_x|>m \quad (or) \quad |b\_y-a\_y|>m$$

Formula 1. Repositioning detector

Where b_x,b_y,a_x and a_y are coordinates of two sampled points. “m” is the minimum distance of two points in each coordinate that satisfy the break point condition. Based on primary sampling, “m” is defined as 2, 3, 4, 7, 10 pixels. For high level tremors, m needs to have small values i.e. 2, 3. The amount 4 is achieved from the user with medial level of tremor. For low level tremors, values of 7 to 10 is enough. At the initializing stage, by default “m” is set at 4 for RBPD. Every second, the number of detected break points will be counted. Later “m” will be reset for RBPD based on detected break points count, If the detected points is more than ten, “m” will be set to a smaller amount as this indicates a user with high-level tremor. The monitoring process will run at all times in parallel mode with RBPD so that the value of “m” is always updated. In other words the RBPD will adapt automatically to the user’s tremor level.

4. Testing and Evaluation

To investigate the effectiveness of RBPD, an experimental test with five (A1-A5) PD patients was conducted. The tremor level for participants, respectively, was: A2 Advanced, A1, A3, A4 Medial and A5 Early Stage. The testing was done at a hospital. Participants were briefed about the experiment and the objectives explained. Participants also did the trial testing. During testing, the participants were asked to follow one pre-defined path on the computer screen by using the mouse. To compare the results from RBPD with fixed amount of “m”, the “m” was set to 2, 3, 4, 7, 10 pixels.

4.1. Results and Discussions

The number of detected break points in each mode is presented in Table 2.

Table 2. Experimental result and detected break points

m	2	3	4	7	10	RBPD
A1	235	198	116	69	52	318
A2	848	610	469	284	223	903
A3	38	244	21	114	81	298
A4	194	135	302	77	134	385
A5	108	74	61	30	165	283

The results show that the participants with a higher level of tremor had more break points in their mouse trajectory. The mean and standard deviation of each “m” and RBPD is presented in Table 3. The descriptive statistics show that the mean for RBPD is higher than the mean of other methods.

Table 3. Descriptive statistic of testing by different values for “m”

”*m*”	Number of participants	Mean	Standard Deviation
2	5	284.600	324.07
3	5	252.200	210.06
4	5	193.800	187.74
7	5	114.800	99.18
10	5	137.000	72.68
RBPD	5	437.000	263.18

Using non-parametric analysis, the results showed that there is a significant difference between different methods (Kendall’s W Test = .60, $p < .01$). This was also true with the Friedman Test ($x2 (5) = 15.06$, $p < .01$). As the mean for RBPD was higher than other methods, this method can be concluded to be much better than the previous ones. The Wilcoxon test (a two related sample non-parametric test) for the RBPD method with each of the other methods showed that the mean of RBPD is significantly higher than other methods.

5. Conclusion

As results show, RBPD had good performance for detecting unsmoothed points in mouse trajectory. Its adaptability helps to get more information from mouse trajectory for further analysis. In the future, the RBPD technique will be bundled with an assistive mouse controlling technique. This assistive technique will try to filter detected break points to provide a smoother mouse trajectory for users with PDs.

Acknowledgement:

The authors would like to thank UKM for funding this research under grant UKM-GUP-2011-235.

References

[1] RJ Elble, and WC Koller. Tremor: Johns Hopkins University Press Baltimore. 1990.

[2] J Accot, and S Zhai. Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. 466-472.

[3] B Johanson, G Hutchins, T Winograd et al. Point Right: experience with flexible input redirection in interactive workspaces. 227-234.

[4] F Hwang, S Keates, P Langdon et al. Multiple haptic targets for motion-impaired computer users. 41-48.

[5] H Hassan. Common input devices for Malaysian computer users with motor impairments. *ACM SIGACCESS Accessibility and Computing*. 2009; (93): 18-25.

[6] JB Freeman and N Ambady. Mouse Tracker: Software for studying real-time mental processing using a computer mouse-tracking method. *Behavior Research Methods*. 2010; 42(1): 226-241.

[7] H Jimison, N Jessey, J McKanna et al. *Monitoring Computer Interactions to Detect Early Cognitive Impairment in Elders*. 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare. 2006: 75-78.