Multidirectional Pointing Input Using a Hardware Keyboard

Byungjoo Lee, Haesun Park, and Hyunwoo Bang

Herein, we propose a novel method to create multidirectional and multilevel pointing input using only a hardware keyboard. The relative position vectors among pressed keys are calculated into an input vector. The pointing performance of this method is evaluated based on the ISO 9241-9 recommendations. We find that the performance of the method is sufficient to create moderate pointing input, as the measurement of the throughput is close to that of a touchpad, as measured in past studies also using the ISO 9241-9 recommendations.

Keywords: Pointing input, hardware keyboard, ISO 9241-9.

I. Introduction

When conducting various tasks on a graphical user interface (GUI)-based operating system, the user most often alternates between a hardware keyboard and a pointing input device, such as a mouse. In such a situation, there is some inconvenience because the user should continually move a hand from the keyboard to the pointing input device and back. Moreover, in a task primarily involving text, although such cases requiring a high level of accuracy are few, the user should move their hand to the pointing input device for small and simple movements [1]-[3]. This frequently causes fatigue of the hand or fingers [4].

The pointing stick is known to resolve the inconvenience of constantly moving the hands [4]. Nevertheless, although the inconvenience can be mitigated, using this device leads to a high level of fatigue in the user's forearm [4], as the user should still apply force with a certain finger constantly. Also, according to one study [5], use of the pointing stick has declined because it is far more difficult to conduct precise movement with the pointing stick than with the track ball or the touchpad. Moreover, this device requires zeroing periodically due to the aging of the sensor. Recently, attempts were made to overcome this inconvenience by means of vision-based systems, with one such system developed to track the user's hands on the keyboard as a type of pointing input [5]. However, this also led to several problems, including the dots per inch level being lower due to the limited image resolution, higher costs, and the need for convoluted algorithms for image post-processing, all of which reduced the processing speed. These products are also sensitive to the surrounding environment [5]. Another study changed the normal surface of the keyboard to a touch-sensitive display [2]. While they succeeded in showing the intriguing possibilities of using the keyboard for graphical manipulation, they did not focus on typical pointing tasks. In addition, their hardware was quite complicated and potentially too fragile to endure the repetitive force applied during typing tasks. Other work [3] combined face and speech recognition to support multimodal pointing input in the local area of a digital screen. While the system was comparable in terms of performance to the use of a mouse-keyboard strategy, speech recognition is generally not acceptable in the average workplace. Lastly, there is a pointing input section on a typical keyboard consisting of up, down, left, and right. This four-directional pointing input system limits the possible directions of input, and the gain is not adaptable during movements. Moreover, the pointing input actions are not conducted in the text input area of the keyboard and therefore involve the overuse of hand movements, as mentioned above.

This study aims to develop a system that can be used as a multidirectional, multilevel pointing input device in the form of a hardware keyboard, which is indispensable for text input.

Manuscript received Mar. 11, 2013; revised May 22, 2013; accepted June 4, 2013.
Byungjoo Lee (phone: +82 2 880 1723, bjlee@snu.ac.kr), Haesun Park (gotjs514@snu.ac.kr), and Hyunwoo Bang (corresponding author, savoy@snu.ac.kr) are with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.13.0213.0117

Consequentially, we expect to minimize the number of hand movements required when alternating between text and simple pointing input. Without other devices attached to a computer, we make it possible with this system to use the position vectors of pressed keys. Finally, we evaluate the feasibility and the performance of the system according to Fitts' law.

II. Operating Principle of System

A flow diagram of the system is depicted in Fig. 1. There are two separate modes in this system. The user can switch between the two modes using a "hotkey" or a "reserved" button on the keyboard. This conversion is completed without the loss of time because the user's hands are already on the keyboard. If the user selects the typical text-input mode, the keyboard receives a command and performs as a normal text-input device. Once the pointing mode is selected, however, the system detects which keys are currently pressed by the user and then saves the identification number of the pressed keys. The position information of the pressed keys is then retrieved from the previously obtained figure data of the keyboard and is used to yield pointing input with respect to a reference point. The figurative data of the keyboard obtained beforehand consists of the relative coordinate information of each key; this data for other keyboards with different form factors should be measured and saved in advance. However, typical shapes of keyboards do not vary much; therefore, once the data is measured, it can most likely be used with another keyboard. We obtain the relative coordinate data by taking a picture of the keyboard from above and marking the center of each key.

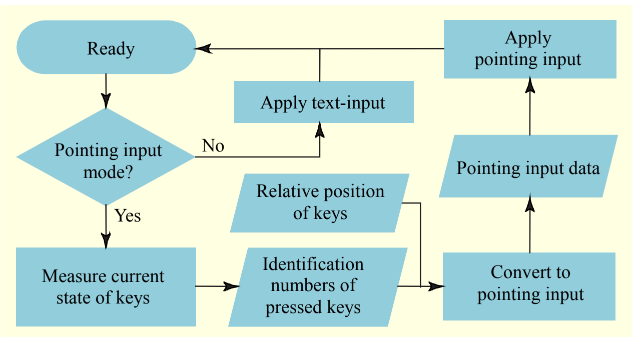

Fig. 1. Flow diagram of system.

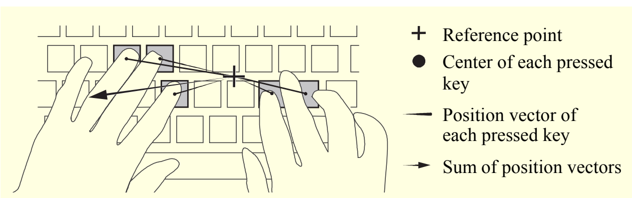

Fig. 2. Estimation of user's directional intention.

The pointing input is estimated by the vector sum of the pressed keys from a reference point. The reference point is located between the "U" key and the "J" key, as depicted in Fig. 2. The location of the reference point is slightly shifted to the left from the center of the keyboard, considering the base location of the user's hand when using the keyboard. We assume the position vector of the *i*-th pressed key to be C_i as well as that of the reference point P. In addition, n is the number of pressed keys at the current time, and G is a constant gain to adjust the sensitivity of the method. We initially measure the typical speed of a cursor controlled by a joystick (up to 10 cm/s) and then select a G value that moves the cursor in the middle of the speed range. In this study, we set G to 20, implying that the slowest possible speed of the cursor is 3.35 cm/s on the display. This value is maintained for all experiments. The input vector $\boldsymbol{v}$ can then be formulated as

$$v=\frac{(C_1-P)+(C_2-P)+\times\times\times+(C_n-P)}{n}\times\frac{1}{G}=\frac{1}{nG}\sum_{i=1}^{n}(C_i-P). \quad (1)$$

This input vector is then treated as a velocity vector when moving the cursor from its current position within a time step.

III. Examining Performance Based on Fitts' Law

The pointing task, which is measured and predicted according to Fitts' law, is one of the most fundamental types of graphical input [6]. Fitts' law is a robust model of human psychomotor behavior that was developed in 1954 [7] for predicting human responses based on rapid targeted movements. It is widely applied to evaluate the performance levels of pointing devices. There have been several formulations of indexes of difficulty in Fitts' law. We use Shannon's formula, which provides a good fit to the data [8].

$$ID = \log_2\left(\frac{D}{W}+1\right), \quad (2)$$

$$MT = a + b\times ID. \quad (3)$$

Here, MT is the movement completion time. ID is the index of difficulty, whose unit is bits. The ID is calculated from D, the distance from the target, and W, the width of the target, both defined in pixels. Properties a and b are estimated by fitting the formula into the empirical data. The throughput is defined as a reciprocal number of b in bits/s.

1. Participants

Eight paid participants are recruited from a local university. The participants' ages range from 20 years old to 28 years old (mean [M] = 24 years old, standard deviation [SD] = 2.62 years old). All participants report that they use computers on a daily basis, and their usage times per day range from 1 h to 8 h (M = 4.33 h, SD = 1.94 h). All participants have normal usability of their hands, and none previously experienced a system or device similar to those used in the experiment.

2. Apparatus

The hardware keyboard, which serves as the input device for the task, is model number RK713A from HP. The task is performed on a 3.30-GHz desktop computer with a 23-inch LED monitor. The resolution of the monitor is 1920×1080. The experimental system is built with Java, with the cursor being a standard arrowhead pointing to the upper left. Although we use a wireless keyboard for this experiment, no noticeable lag is noted while operating it.

3. Task

We implement the multidirectional pointing task specified in the ISO 9241-9 standard [9]. The task is to point between a red circle and a blue circle that are simultaneously displayed on a black background (see Fig. 3). The participant is asked to move the cursor to the inside of the red starting circle and to push the spacebar to begin a trial. Then, immediately after the red circle disappears, the participant is required to move the cursor to the inside of the blue target circle and push the spacebar to end the current trial. The two circles are located symmetrically relative to the center point of the screen. The width of the target, the distance to the target, and the angle of approach are varied, with 3 (15 pixels, 25 pixels, and 35 pixels), 2 (400 pixels and 800 pixels) and 8 (0, $\pm\pi/4$, $\pm\pi/2$, $\pm3\pi/4$, and π radians) levels, respectively.

4. Designs and Procedure

The participant is seated such that they are aligned with the midline of the computer screen and are free to utilize the entire workspace to accommodate their various control strategies. Before the experiment, they complete a pretest questionnaire about their age, gender, and computer usage. The participant is instructed to point to the blue target after clicking the red target as quickly as possible. The experiment is a 3×2×8 within-subjects design with repeated measures ANOVA (analysis of variance). An alpha level of 0.05 is used for all tests. The experiment consists of five blocks, resulting in a total of 240 trials for each participant. Before starting the trial, one practice block is given. Between two blocks, the participant is given a two-minute break. For each block, the independent variables are randomly presented to the participant.

IV. Results and Discussion

We consider a learning effect that shows a significant difference in the trial completion time for the block: $F(4, 28) = 6.80$ and p=0.001. The Helmert contrast shows that the effect becomes insignificant (p=0.168) after the second block. Therefore, we exclude the first block and average all subsequent blocks to obtain the throughput of the method. The resulting throughput of the method is 0.95 bits/s (see Fig. 4), and the correlation of the fitting is 0.99 (1.1 bits/s and 0.917 with effective width compensation). The M of the error rate is 6.1% (SD=1.61%), which is comparable to the results of past studies [8].

Although the performance of the system is not as high as when using a mouse (approximately 3 bits/s to 4 bits/s) [8], the performance of the system is comparable to the throughput of a touchpad, according to results obtained in earlier research (0.99 bits/s to 1.43 bits/s) [10]. This indicates that this method is sufficient to realize simple pointing and moving input actions, such as page scrolling, icon selecting, and clicking, without moving the hand to another device while the user is engaged in a text-dominant task. Another interesting study reported a large number of submovements associated with the touchpad [11] resulting from an excessive amount of clutching action.

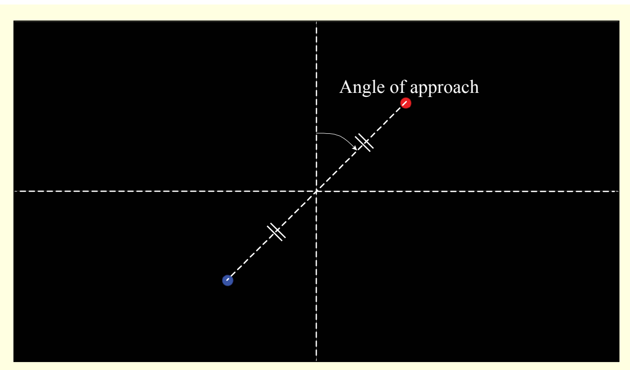

Fig. 3. Screenshot of trial during task.

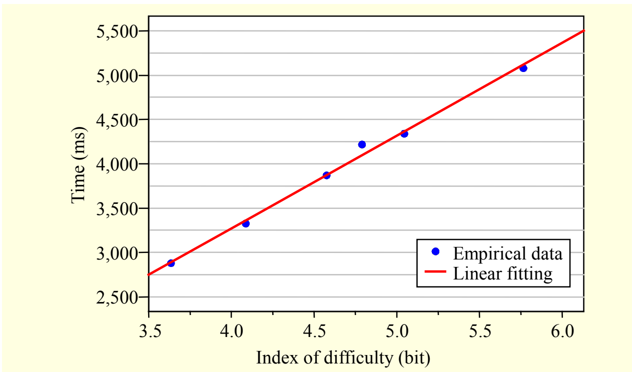

Fig. 4. Empirical data fitted to Fitts' law model.

ETRI Journal, Volume 35, Number 6, December 2013
http://dx.doi.org/10.4218/etrij.13.0213.0117

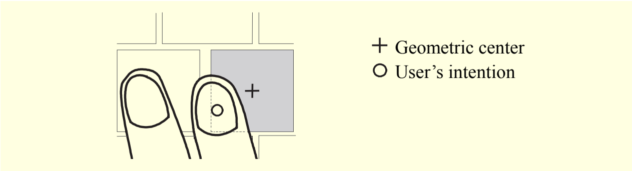

Fig. 5. Calibrating user intentions based on human factors.

However, our method does not involve a large amount of hand movement, making it capable of efficiently achieving appropriate throughput.

We note that our system can be used to manipulate a laptop cursor at a low cost. In general, cursor manipulation is imperfect when using only a touchpad. A touchpad occasionally must be augmented with a mouse to compensate for its inefficiencies when performing precise tasks [12]. Moreover, as our system requires almost no additional hardware to implement, it is instantly applicable to existing devices. This advantage may be useful when there is no need for precise control compared to the complexity required to achieve it.

However, this system can still be improved in that it has a high level of error when attempting minute control during the final stage of a pointing task (the correction phase), as the key arrangement on the keyboard is slightly tilted for typing. Therefore, pinpoint horizontal or vertical movement control is laborious. Moreover, the number of available directions is reduced near the reference point. Tilting the coordinates of the keys is one option that may solve these problems, or we can reverse the mapping by assigning smaller input gains for more distant keys so as to make more directions available for users. Also, to make this system more precise, we can use the time information of key-pressing events, such as approximating the velocity of hand movements to obtain more precise control of intended directions. In addition, based on human factors related to the use of a keyboard, we can use different representatives for each key other than merely their center points (see Fig. 5).

Finally, it is interesting that the performance of this technique is also sufficiently modeled by Fitts' law. Any pointing input manipulated by a user is divided into small units of movement or submovements that do not rely on user feedback control [13]. Regarding this fact, our technique and the manner of controlling the mouse do not appear to differ from each other. In addition, the key rollover problem does not become crucial, which represents a limitation considering the number of keys that one can press simultaneously when using a hardware keyboard. In our study, this limit is five, and it is sufficient to manipulate short and linear submovements.

V. Conclusion

This research aimed to develop a system with a graphical user interface that allows a user to create pointing input only with a keyboard. The pointing input is estimated through the vector sum of pressed keys with respect to a reference point. The performance of the system was evaluated according to Fitts' law, and the result showed that the system has throughput comparable to that of a touchpad.

References

[1] J. Kato, S. Daisuke, and T. Igarashi, "Surfboard: Keyboard with Microphone as a Low-Cost Interactive Surface," *Adjunct Proc. 23rd Annual ACM Symp. User Interface Software Technol*., Oct. 2010, pp. 387-388.

[2] F. Block, G. Hans, and N. Villar, "Touch-Display Keyboards: Transforming Keyboards into Interactive Surfaces," *Proc. 28th Int. Conf. Human Factors Comput. Syst.*, Apr. 2010, pp. 1145-1154.

[3] J. Na, W. Choi, and D. Lee, "Design and Implementation of a Multimodal Input Device Using a Web Camera," *ETRI J.*, vol. 30, no. 4, Aug. 2008, pp. 621-623.

[4] E. Fernström and M.O. Ericson. "Computer Mouse or Trackpoint — Effects on Muscular Load and Operator Experience," *Appl. Ergonomics*, vol. 28, no. 5, 1997, pp. 347-354.

[5] A. Wilson and E. Cutrell, "Flowmouse: A Computer Vision-Based Pointing and Gesture Input Device," *Human-Comput. Interaction-INTERACT*, 2005, pp. 565-578.

[6] G. Casiez et al., "The Impact of Control-Display Gain on User Performance in Pointing Tasks," *Human-Comput. Interaction*, vol. 23, no. 3, 2008, pp. 215-250.

[7] P.M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *J. Experimental Psychology*, vol. 47, no. 6, 1954, p. 381.

[8] R.W. Soukoreff and I.S. MacKenzie, "Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI," *Int. J. Human-Comput. Studies*, vol. 61, no. 6, 2004, pp. 751-789.

[9] ISO Std. 9241-9, *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) — Part 9: Requirements for Non-keyboard Input Devices*, International Organization for Standardization, 2000.

[10] I.S. MacKenzie and A. Oniszczak, "A Comparison of Three Selection Techniques for Touchpads," *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, vol. 18, no. 23, 1998, pp. 336-343.

[11] H. Dillen, J.S. Farris, and J.W. Meehan, "Kinematic Analysis of Cursor Trajectories Controlled with a Touchpad," *Int. J. Human-Comput. Interaction*, vol. 19, no. 2, 2005, pp. 223-239.

[12] S.K. Tang et al., "Virtual Mouse: A Low Cost Proximity-Based Gestural Pointing Device," *LNCS*, vol. 6762, 2011, pp. 491-499.

[13] D.E. Meyer et al., "Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements," *Psychological Rev.*, vol. 95, no. 3, 1988, pp. 340-370.