Ex-Space: Expanded Space Key by Sliding Thumb on Home Position

Kodai Sekimori(✉), Yusuke Yamasaki, Yuki Takagi, Kazuma Murata, Buntarou Shizuki, and Shin Takahashi

University of Tsukuba, Tsukuba, Japan
{sekimori,yusukeyamasaki,takagi,kmurata}@iplab.cs.tsukuba.ac.jp,
{shizuki,shin}@cs.tsukuba.ac.jp

Abstract. We describe a method termed Ex-Space to enable continuous, one-dimensional keyboard input on the home position. The method features thumb movement on the space key. We developed a prototype of the method consisting of a piece of conductive fabric attached to the space key to detect the position of the sliding thumb. In addition, we performed two preliminary experiments: an experiment to calculate the resolution afforded by touch position detection, and an experiment to explore the performance of menu command input using Ex-Space.

Keywords: Touch · Keyboard · Menu · Home position · Space key

1 Introduction

Graphical user interface (GUI) menus are used to input commands in many applications, such as text editors and browsers. They support 'see and select'-style inputs; users first view menu items and then select one using a pointing device, such as a mouse. Therefore, if the application also requires keyboard input, users have to move their hands between the mouse and keyboard. This 'homing' operation requires 0.4 s on average [1,2], which is considerable. One way to alleviate the problem is to assign a shortcut key to each command in the menu. However, it is then difficult to remember the large number of key assignments [6,7].

Here, we propose a novel input method termed 'Ex-Space'; we use a customized keyboard to sense user touch on the surface of the space key. The user can move the cursor through menu items, and then select the required item by sliding the thumb on the space key (Fig. 1). During these operations, the hand remains on the home position of the keyboard; there is no need to move the hand to (for example) a mouse.

In this paper, we describe the Ex-Space input method and its prototype implementation, where touch position detection is enabled by attaching conductive fabric to the space key. We also show two preliminary experiments: an experiment to calculate the resolution afforded by touch position detection, and an experiment to measure the performance of menu command input using Ex-Space.

M. Kurosu (Ed.): HCI 2018, LNCS 10903, pp. 68–78, 2018.
https://doi.org/10.1007/978-3-319-91250-9_6

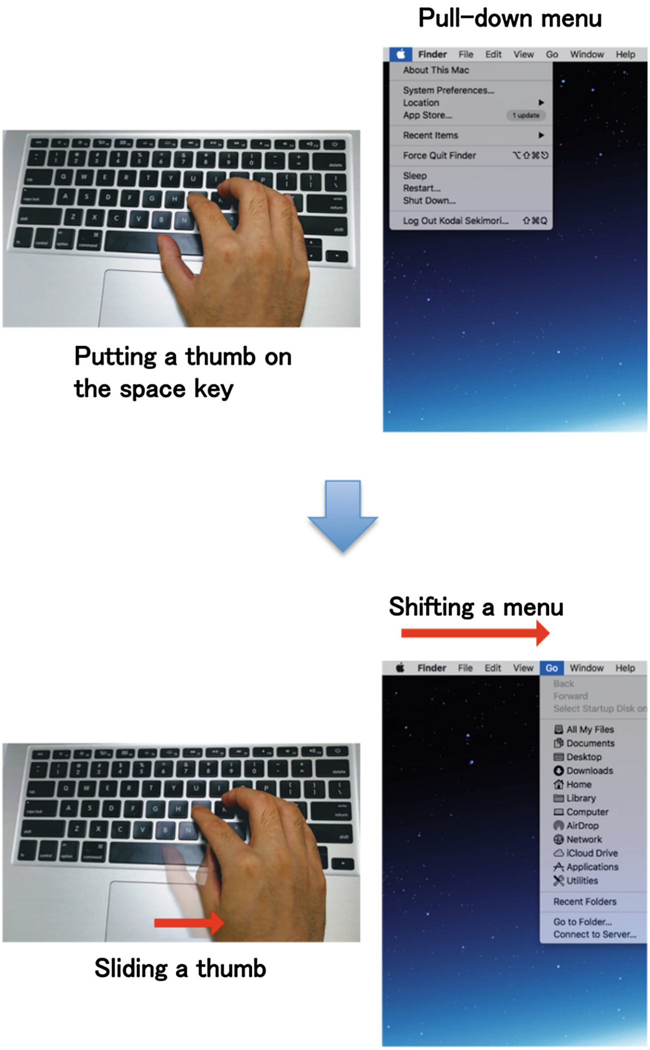

Fig. 1. An image of Ex-Space in action; when the user slides the thumb to the right on the space key, the pull-down menu shifts to the right.

2 Related Work

Some keyboards allow advanced user interactions. ThinkPad[1] is equipped with a pointer; the user can point without moving their hand from the home position of the keyboard. The most recent version of the MacBook Pro has a Touch Bar[2] on top of the keyboard, which is an interactive touch display that changes the

[1] https://support.lenovo.com/us/en/solutions/pd026744.
[2] https://www.apple.com/macbook-pro/.

function according to the application being executed. The keyboard of Poseidon Z Touch[3] enables touch interaction via sensors attached to the space key.

Several devices that do not require the 'homing' operation have been developed. Rekimoto [8] developed a method featuring a combined keyboard/touchpad; the thumb was moved on the touchpad with the hands on the keyboard. The GestKeyboard [11] detects hand gestures by using the intervals between the pressing of several keys on an ordinary keyboard.

Many methods of detecting hand gestures above keyboards equipped with sensors are available. The silicon-covered FlickBoard [10] incorporates a capacitative grid that recognizes hand gestures. Type-Hover-Swipe [9] recognizes hand gestures using 64 photo reflectors interspersed between the keys of the keyboard. Keyboard Clawing [5] detects clawing gestures using acoustic sensing. Surfboard [4] detects simple gestures, such as horizontal hand movement above the keyboard, using a microphone attached to the keyboard. Dietz et al. [3] proposed a method in which pressure sensors embedded under each key sense the force with which the keys are pressed.

3 Ex-Space

Ex-Space is a novel method allowing selection of GUI menu items by sliding the thumb on the space key. Because the space key is long, a continuous one-dimensional input (i.e., 'how much the thumb moves on the space key') is feasible. It then becomes possible to perform interactions without 'homing'.

GUI menu operation is performed by pressing a key (hereinafter referred to as the TRGKEY) that triggers the operation. Then, the GUI menu is operated by thumb sliding. By pressing TRGKEY during the thumb slide, the GUI menu operation is canceled. In our Ex-Space prototype, TRGKEY is the control (Ctrl) key.

We now describe the detailed operation of Ex-Space. The features of the GUI menu are shown in Fig. 2. Ex-Space is a two-step selection method; a menu is selected in the menu bar and a menu item from the selected menu. When selecting a menu in the menu bar, the index of the menu to be selected moves to the right when the thumb slides to the right on the space key, and to the left when the thumb slides to the left. When selecting a menu item, the index of the menu item to be selected moves downward when the thumb slides to the right, and upward when the thumb slides to the left. Regardless of the number of menus and menu items, the leftmost menu and the top menu item are associated with the left end of the space key, and the rightmost menu and the bottom menu item with the right end (Fig. 3). Thus, Ex-Space selects all menus and all menu items in the same manner.

There are five steps in the selection of a menu item.

1. Enter the menu selection state of the menu bar by pressing TRGKEY.
2. Select the required menu by sliding the thumb.

[3] http://www.ttesports.com/productPage.aspx?p=209&g=ftr.

3. Open the selected menu and enter the menu item selection state by releasing the thumb.
4. Select the menu item by sliding the thumb.
5. Open the selected menu item by releasing the thumb.

If the menu contains a submenu, that submenu is opened by selection of the menu item, followed by a transition to selection of the menu item in the submenu. The submenu item is selected by removing the thumb from the space key, as described above for selection of an item in a menu.

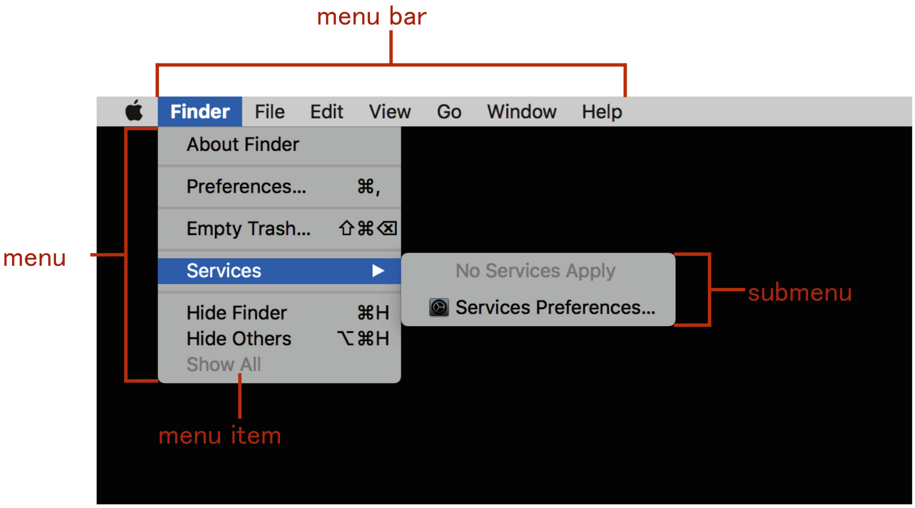

Fig. 2. The components of the graphical user interface (GUI) menu.

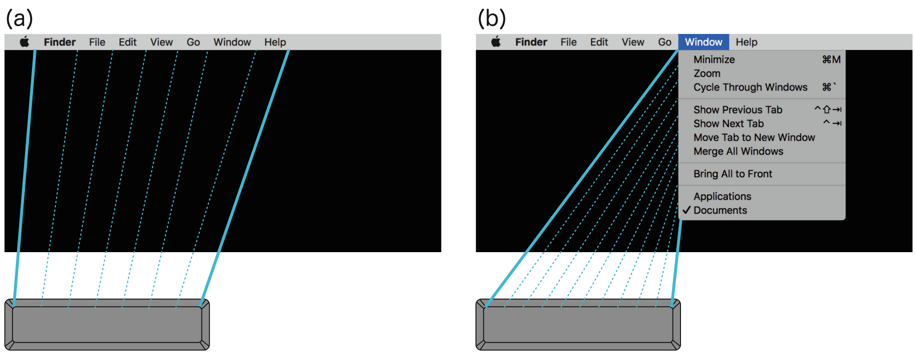

Fig. 3. Space key is associated with (a) the menu in the menu bar and (b) the menu item.

4 Prototype Implementation

We created an Ex-Space prototype detecting sliding thumb movement on the space key. A schematic view of the prototype is shown in Fig. 4. EeonTex Pressure Sensing Fabric[4], a conductive fabric, was attached to the space key. Two drive electrodes were attached to one end, and 200 kHz sinusoidal alternating current generated by a function generator flowed between the electrodes. The current created an electric field in the conductive fabric, and the shunt current varied by the position of the finger/thumb on the fabric. The voltage was measured by another electrode (the measuring electrode) attached to the other end of the space key; the closer the touch position was to the measuring electrode, the higher the voltage. By measuring the change in voltage, the touch position could be detected. To expand the dynamic range of A/D conversion, the measured signal was amplified by two non-inverting amplifiers, passed through a low-pass filter to remove the AC component of the current, and sampled by the A/D converter of the Arduino MEGA microcontroller[5].

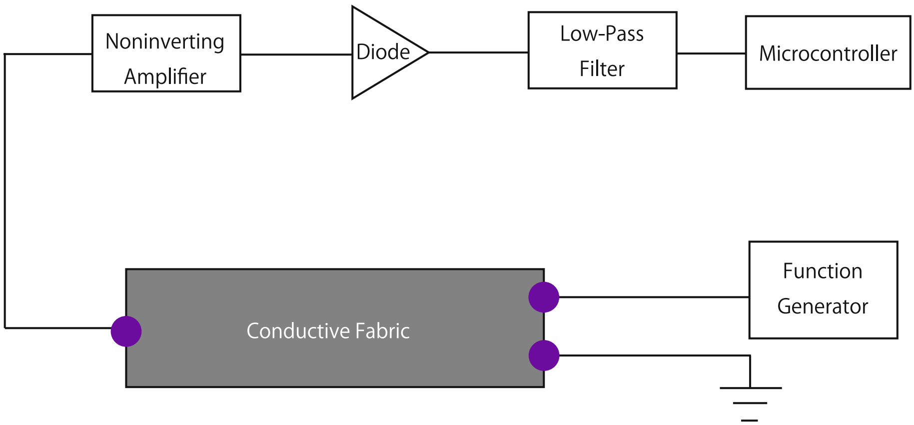

Fig. 4. A schematic of Ex-Space.

5 Preliminary Experiment 1

We explored the relationship between the position touched by the thumb and the voltage, to define the resolution of touch detection.

5.1 Experimental Design

We removed the space key from the keyboard, attached it to another board (Fig. 5), and attached a piece of conductive fabric to the space key. We then

[4] https://www.sparkfun.com/products/14111.
[5] https://store.arduino.cc/usa/arduino-mega-2560-rev3.

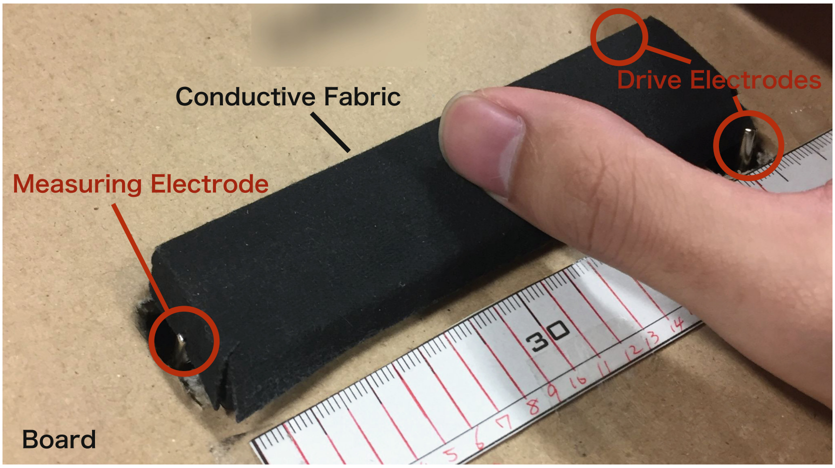

Fig. 5. Experiment situation: the space key was removed from the keyboard and attached to the board, and a conductive piece of fabric was attached to the space key for this experiment.

attached two electrodes at the right end, and another at the left end, of the space key. An AC signal flows between the two electrodes at the right end, and the voltage is measured by the electrode at the left end. Five volunteers participated in the experiment. They sequentially touched the conductive fabric with the right thumb, from the left to the right end, in a total of 20 places according to a spacing of 5 mm. We recorded the voltage at each position 100 times for each participant. The space key is about 110 mm in length.

5.2 Results

Figure 6 shows the results of the experiment. The X-axis represents the distance from the drive electrodes (the two electrodes on the right) and the Y-axis denotes the voltage. Each voltage is the average of 100 measurements.

The voltage increased linearly to 70 mm, but saturation developed thereafter. Therefore, the touch position can be estimated accurately to the point 70 mm from the right end. Saturation is explained by the fact that as the distance away from the drive electrodes increases, the amount of shunt current decreases, and thus the influence thereof on the electric field also falls.

6 Preliminary Experiment 2

We explored the performance of menu command input using Ex-Space.

6.1 Participants

We enrolled 4 male participants (P1–P4, 21–24 years of age; right-handed). All participants usually use PCs.

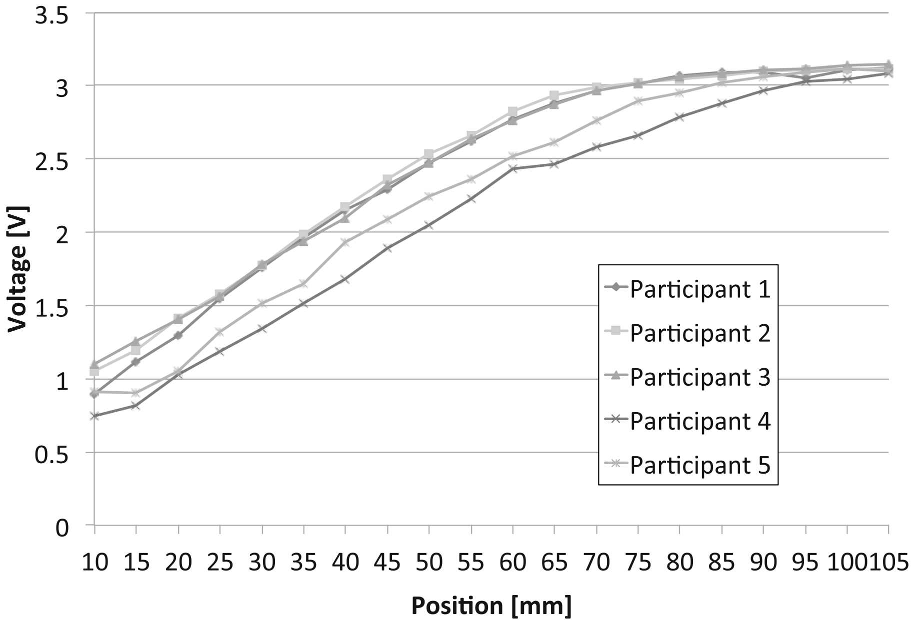

Fig. 6. Position-voltage curve

6.2 Devices and Software

We developed recognition software based on the previous experiment to detect the thumb touch position. Since the conductive fabric used in the previous experiment was affected by the touch pressure, EeonTex Conductive Stretchable Fabric[6], which was not affected by pressure was attached to the space key in this experiment. In addition, the AC signal voltage was increased to alleviate the saturation problem.

We implemented the experimental application using Python (the PyQt5[7] library handles GUI). The menu of the application is shown in Fig. 7. The application has 8 menus, and each menu contains 8 menu items. Each participant selected menus using Ex-Space. The application reveals one menu item (the target item) to the participant, and the participant selects that item.

6.3 Procedure and Task

Each participant practiced selecting items using Ex-Space until familiarity was attained. For each trial, the participant selected the menu item that matched the target item displayed on the application. One session consisted of 10 trials and each participant performed 2 sessions. Therefore, the total number of trials

[6] https://www.sparkfun.com/products/14112.
[7] https://www.riverbankcomputing.com/software/pyqt/intro.

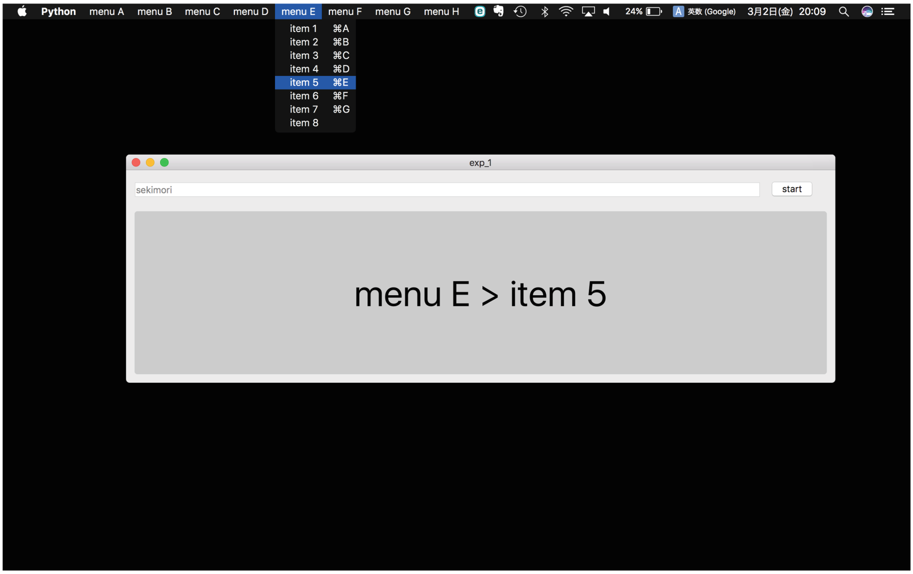

Fig. 7. Experimental application; each participant selects items on the menu displayed in the application.

was as follows: 10 trials $\times$ 2 sessions = 20 trials per participant. The application measures the time that the participant takes to select the item, even if the selected item did not match the target item. When a participant finished selection of a menu item and pressed the 'j' key, the application showed the next target item. The application calculates the average selection times and correct rates that the selected item matches the target item.

6.4 Results and Discussion

The average times taken to select menu items and the correct answer rates are shown in Figs. 8 and 9. The average selection time was 13.88 s (SD = 11.24). The average correct answer rate was 30.0%.

The performance of menu command input using Ex-Space was not so fast and accurate. To investigate the reason of this result, we analyzed the logs of users' trials, and found that the users often mistakenly selected items adjacent to the correct target, that is, the touch position detection is still unreliable. One of the reason for this is that various electrical noises are contaminated through the user's body when he/she touches the space key. This problem should be solved by using noise cancellation techniques such as band-pass filters, or by increasing the number of measuring electrodes. Decreasing the number of menu items for the space key will also alleviate the problem.

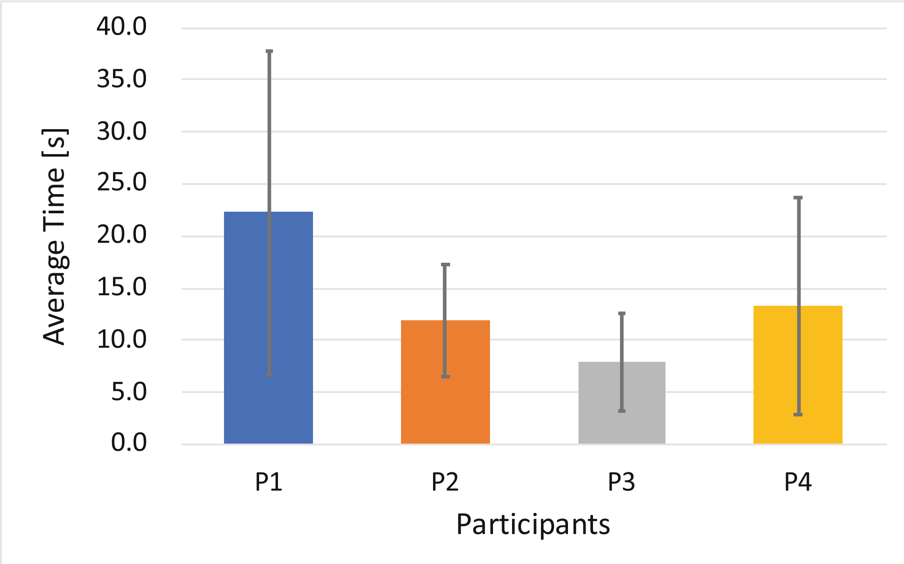

Fig. 8. The average selection time of each participant.

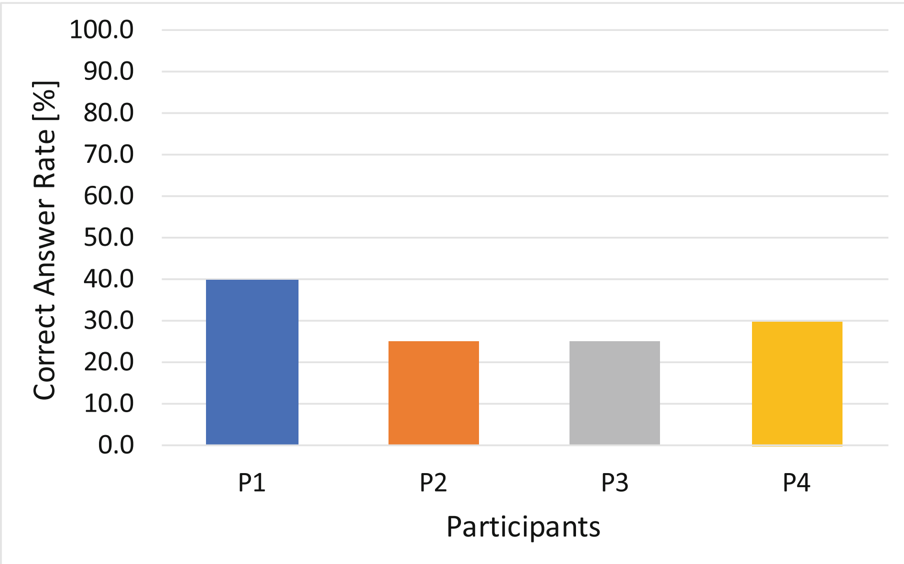

Fig. 9. The correct answer rates of each participant.

7 Conclusion

We developed the Ex-Space method to enable continuous one-dimensional input on the home position, using thumb movement on the space key as input. We attached conductive fabric to the space key and detected touch and slide gestures by measuring changes in the electric field depending on where the thumb touched the surface. We then developed a prototype of the method to detect the position of the sliding thumb. We explored the performance of menu command input using Ex-Space. The performance was still not so fast and accurate. Due to the problem of electrical noise contamination, the users often mistakenly selected items adjacent to the correct target. This problem should be solved by the noise cancelation or decreasing the number of menu items. In future work, we will improve the implementation of touch position detection and explore the performance of the Ex-Space input method.

Acknowledgements. We would like to thank Professor Simona Vasilache and Pedro Passos Couteiro for their great advices to our paper.

References

1. Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. Commun. ACM **23**(7), 396–410 (1980). https://doi.org/10.1145/358886.358895
2. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale (1983)
3. Dietz, P.H., Eidelson, B., Westhues, J., Bathiche, S.: A practical pressure sensitive computer keyboard. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, UIST 2009, pp. 55–58. ACM, New York (2009). https://doi.org/10.1145/1622176.1622187
4. Kato, J., Sakamoto, D., Igarashi, T.: Surfboard: keyboard with microphone as a low-cost interactive surface. In: Adjunct Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology, UIST 2010, pp. 387–388. ACM, New York (2010). https://doi.org/10.1145/1866218.1866233
5. Kurosawa, T., Shizuki, B., Tanaka, J.: Keyboard clawing: input method by clawing key tops. In: Kurosu, M. (ed.) HCI 2013. LNCS, vol. 8007, pp. 272–280. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39330-3_29
6. Lane, D.M., Napier, H.A., Peres, S.C., Sándor, A.: Hidden costs of graphical user interfaces: failure to make the transition from menus and icon toolbars to keyboard shortcuts. Int. J. Hum.-Comput. Interact. **18**(2), 133–144 (2005)
7. Malacria, S., Bailly, G., Harrison, J., Cockburn, A., Gutwin, C.: Promoting hotkey use through rehearsal with ExposeHK. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 573–582. ACM, New York (2013). https://doi.org/10.1145/2470654.2470735
8. Rekimoto, J.: ThumbSense: automatic input mode sensing for touchpad-based interactions. In: CHI 2003 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2003, pp. 852–853. ACM, New York (2003). https://doi.org/10.1145/765891.766031

9. Taylor, S., Keskin, C., Hilliges, O., Izadi, S., Helmes, J.: Type-hover-swipe in 96 bytes: a motion sensing mechanical keyboard. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI 2014, pp. 1695–1704. ACM, New York (2014). https://doi.org/10.1145/2556288.2557030
10. Tung, Y.C., Cheng, T.Y., Yu, N.H., Wang, C., Chen, M.Y.: FlickBoard: enabling trackpad interaction with automatic mode switching on a capacitive-sensing keyboard. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, pp. 1847–1850. ACM, New York (2015). https://doi.org/10.1145/2702123.2702582
11. Zhang, H., Li, Y.: GestKeyboard: enabling gesture-based interaction on ordinary physical keyboard. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI 2014, pp. 1675–1684. ACM, New York (2014). https://doi.org/10.1145/2556288.2557362