Comparison of Two Methods to Control the Mouse Using a Keypad

Torsten Felzer[1], Ian Scott MacKenzie[2](✉), and John Magee[3]

[1] Institute for Mechatronic Systems, Technische Universität Darmstadt, Darmstadt, Germany
`felzer@ims.tu-darmstadt.de`
[2] Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
`mack@cse.yorku.ca`
[3] Math and Computer Science Department, Clark University, Worcester, MA 01610, USA
`jmagee@clarku.edu`

Abstract. This paper presents a user study comparing two methods for keyboard-driven mouse replacement: CKM, an active Conventional Keyboard Mouse, and DualMouse, an innovative keyboard technique allowing stepwise, recursive target acquisition. Both strategies are implemented in the pointing component of OnScreenDualScribe, a comprehensive assistive software system that turns a compact keypad into a universal input device. The study involves eight non-disabled participants and a single user with Friedreich Ataxia. The results reveal that CKM yields about 60 % higher throughput that DualMouse. However, the DualMouse technique is preferable for certain specific tasks. Our intention with this research is to gain new insights into OnScreenDualScribe and to inspire future developers of mouse-replacement interfaces for persons with physical disabilities.

Keywords: Assistive technology · Neuro-muscular diseases · Keyboard replacement · Mouse replacement · Fitts' law · Real-world use

1 Introduction

Computer users with a motor disability often rely on alternative input interfaces, since standard entry devices (e.g., a keyboard and mouse) might be cumbersome, error-prone, inefficient, effortful, or impossible to use. One example for an alternative interface is OSDS (OnScreenDualScribe) [5] which was designed for persons with certain neuro-muscular diseases. OSDS receives input from a modified numeric keypad called DualPad (Fig. 1).

The main problem with full-size keyboards is the need to frequently reposition the hands between keys [13]. DualPad avoids this, since it is securely grabbed with both hands with every key reached from the same hand position. However, computer interaction also relies on mouse control. Switching between multiple devices would eliminate DualPad's advantage. Such was the drawback of the initial version of OSDS [3], which only replaced the keyboard. The current version implements three methods to control

K. Miesenberger et al. (Eds.): ICCHP 2016, Part II, LNCS 9759, pp. 511–518, 2016.
DOI: 10.1007/978-3-319-41267-2_72

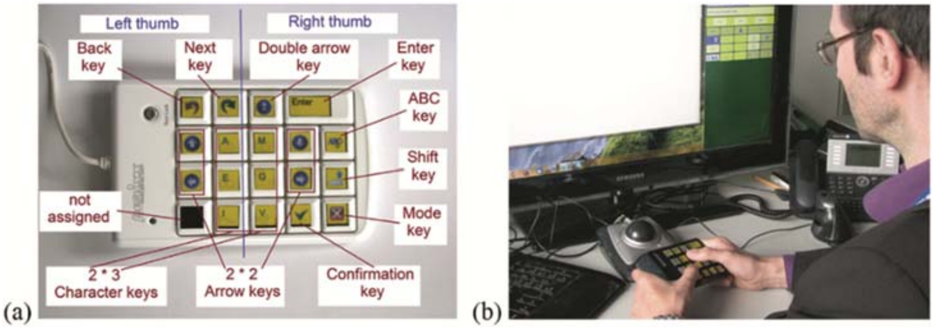

Fig. 1. Input device used in the study: (a) key labels (b) in action

the mouse. The methods are implemented internally and do not require an extra pointing device. Two of those methods are compared in this paper.

2 Related Work

The first author is the creator of OnScreenDualScribe and has previously presented details of its development, evaluation, and related work [3, 6]. The authors previously employed an evaluation technique similar that presented in this paper for mouse selection modalities with a camera-based interface [10]. The evaluation used Fitts' law, which is widely used for pointing device evaluation [9, 14]. Mouse pointing and selection tasks could also be modeled as a Keystroke-Level-Model using the Goals, Objects, Methods, Selection (GOMS) model [2, 8].

A Fitts' law evaluation on the AngleMouse [16] showed improved throughput for users with motor impairments while showing no significant differences for non-disabled users. Similar evaluation techniques were used to compare head orientation against neck muscle EMG signals in a pointer interface [15].

Evaluations of Alternative and Augmentative Communication (AAC) methods in the rehabilitation and speech pathology communities often focus on functional and outcome-based metrics (e.g., [1]). In the HCI and Computer Science communities, participatory design and user satisfaction metrics are often employed (e.g., [11]).

Other relevant point-select evaluations measure both qualitative user feedback and quantitative performance. For example, the Manual and Gaze Input Cascaded (MAGIC) [17] technique warps the mouse pointer to an intended area of the screen for further refinement. A user study found reduced physical effort and fatigue as compared to traditional manual pointing, greater accuracy and naturalness than traditional gaze pointing, and comparable or faster speed than manual pointing.

3 Implementation

To interact with a computer, persons who rely on a compact, tangible interface may use OSDS which serves as a driver for the keypad depicted in Fig. 1. The program computes virtual input events sent to the active window based on physical input from the user.

In this way, it fully emulates a standard keyboard and a two-button mouse (complying to [7] in the context of "real-world" applications).

For mouse emulation, the user chooses among three methods [6]. The first method is a simple active keyboard mouse (also called Continuous Keyboard Mouse or CKM) with certain keys for moving the mouse pointer in cardinal directions and other keys for issuing clicks at the current position. It is "active" because mouse movement continues while the user actively presses a key.

The second method (DKM for Discrete Keyboard Mouse) is similar, but the user only initiates a mouse movement in a certain direction, and then passively waits – while the pointer moves in that direction – before stopping the pointer by pressing a key. Both techniques have the disadvantage of requiring physical input events at specific points in time (so that the movement does not overshoot). This might be problematic for users with deficient fine-motor control. The third strategy, DualMouse, does not rely on mouse movement at all, but directly clicks at a destination location selected by the user following a step-by-step locating process.

The example in Fig. 2 gives an explanation of DualMouse. Suppose, the goal is to click on the little white circle below "Johannes Kepler Universität" in a web browser showing the ICCHP map. Initially, the screen is divided into 24 rectangles or cells (Fig. 2a). The user then selects the row and column of the cell containing the target. This is recursively repeated (Fig. 2b, c and d), meaning the cell is sub-divided and the user selects the appropriate sub-cell. Recursion can terminate (resulting in a click) if the target lies in the center of the current sub-cell (in the example, Fig. 2d). Left-clicking at this destination accesses details on the university hosting ICCHP (Fig. 2e).

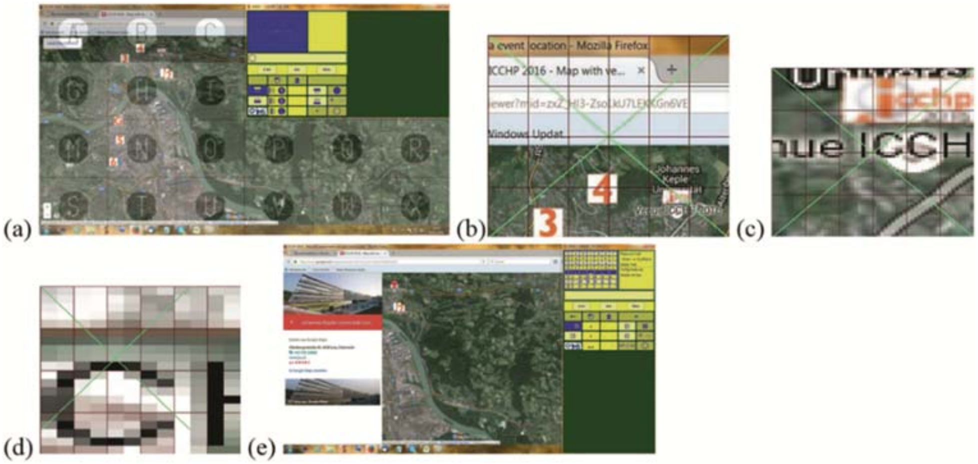

Fig. 2. Stepwise mouse control: (a) screen overlaid with 4 × 6 grid (b) recursion on cell B (c) enlarged sub-cell (d) last necessary refinement (e) screen after click (see text for explanation)

4 Evaluation

This section and the next describe a user study and a case study. First, we describe a user study with eight non-disabled participants (mean age 25 years, 2 female).

The results serve as a baseline for the input methods under test and for comparison with a case study with a representative user.

Two of the input methods described in Sect. 3 were chosen for evaluation: CKM and DualMouse. The evaluation uses FittsTaskTwo which implements the ISO 9241 Part 9 protocol for evaluating pointing devices (see [10] for related work using the same software). The task involves selecting circular targets of a specified width (*W*) at a specified amplitude (*A*) in a certain order. Thirteen target circles are arranged in a layout circle, as shown in Fig. 3.

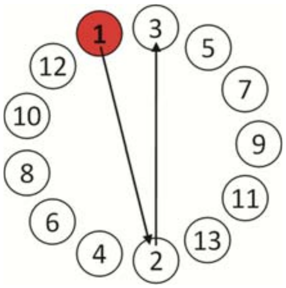

Fig. 3. FittsTaskTwo evaluation software (explanation in text)

Selecting all thirteen targets constitutes a sequence of trials. The diameter of the layout circle sets the amplitude of movement. Two amplitudes were used: 300 pixels and 500 pixels. Two target widths were used: 30 pixels and 60 pixels. Thus, there were 8 participants × 2 input methods × 4 blocks × 2 amplitudes × 2 widths × 13 selections per sequence = 3328 total trials. This is a relatively small amount of testing; however, this was a practical necessity as early pilot tests revealed that the time for each selection is 10 to 20 times longer than the time to do the same task using a regular mouse (with non-disabled users). Each input method was tested in a separate session, which lasted about one hour per participant. The two input methods were counterbalanced (4 participants per group) to offset learning effects.

The case study engaged a user from the target community. The participant is a 45-year-old male computer user, the first author, who has the neuromuscular disease Friedreich Ataxia. Due to the disease, he has deteriorating motor control problems keeping him from efficiently utilizing a full-size keyboard. His voice is dysarthric, so he cannot use speech recognition as an alternative. However, he (presently) has the manual control ability to hold the keypad in both hands and press keys with the thumbs, and thus uses OSDS daily to interact with a computer.

5 Results and Discussion

Although data for several dependent variables were collected, due to space limitations we focus primarily on the results for throughput. Throughput is a composite measure computed from the speed and accuracy in selecting targets [14].

In the user study with non-disabled participants ($n = 8$), the grand mean for throughput was 0.41 bits/s. This value is low – about 1/10th the value typically obtained in mouse studies (see [14] for examples).[1] Clearly, the methods evaluated herein are not competitive with mouse input for non-disabled users. The main limitation is the movement time (*MT*) to select targets. *MTs* were typically in the range of 5 to 20 s per trial. This is compared to *MTs* typically under 1 s for similar tasks performed with a mouse.

The results for throughput by input method are seen in Fig. 4a. Throughputs were 0.31 bits/s for DualMouse and 0.52 bits/for CKM, representing a performance advantage of 68 % for CKM. The difference was statistically significant ($F_{1,6} = 115.4$, $p < .0001$). The results were consistent by participant, with throughputs ranging from 0.41 bits/s (P03) to 0.64 bits/s (P07) for CKM and from 0.19 bits/s (P03) to 0.60 bits/s (P07) for DualMouse. See Fig. 4b. Although not shown, there was a statistically significant improvement over the four blocks of trials ($F_{3,18} = 23.7$, $p < 0001$).

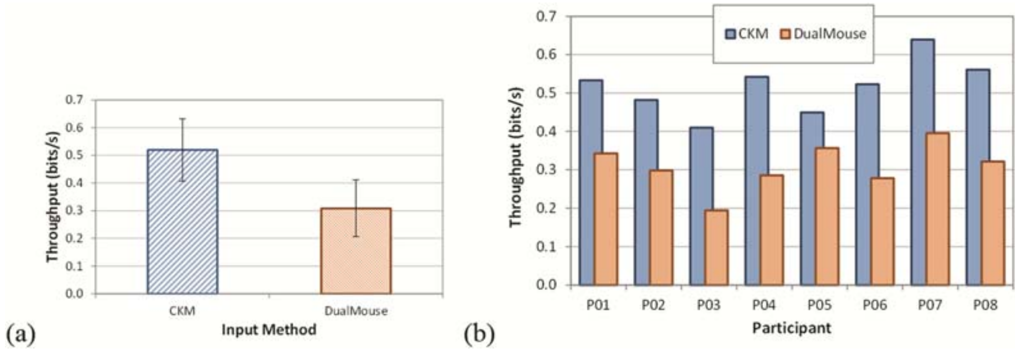

Fig. 4. User study results for throughput (bits/s) (a) by input method and (b) by participant (Color figure online)

In the case study with a user from the target community (n = 1), the grand mean for throughput was 0.17 bit/s. By input method, the means were 0.21 bits/s for CKM and 0.13 bits/s for DualMouse. See Fig. 5. This reflects a performance advantage of 63 % for the CKM. Interestingly, this performance advantage for CKM, as a percent, was similar to the 68 % advantage for CKM in the user study with non-disabled participants.

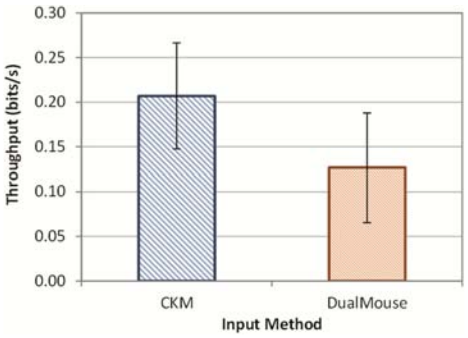

Fig. 5. Case study results for throughput (bits/s) by input method

---

[1] Each participant also did one post-experiment block of trials using a mouse. The mean throughput for the mouse trials was 4.26 bits/s.

Some examples of pointer traces are seen in Fig. 6. The remarkably different patterns are due entirely to the unique ways of repositioning the pointer in the CKM (left) and DualMouse (right) input methods. With the CKM input method, each selection is at a location differing from the last selection only in the x-coordinate (lateral movements) or y-coordinate (vertical movements). Hence, movement appears as lateral or vertical jumps in the cursor position. By contrast, with the DualMouse input method, recursively sub-dividing and zooming in to a location causes each selection to differ from the preceding selection by both the x- and y-coordinate. Hence, there is an apparent pattern of diagonal movement in the cursor position. Importantly, there is cognitive agreement between how the user feels the cursor is moving and the patterns shown in the figure.

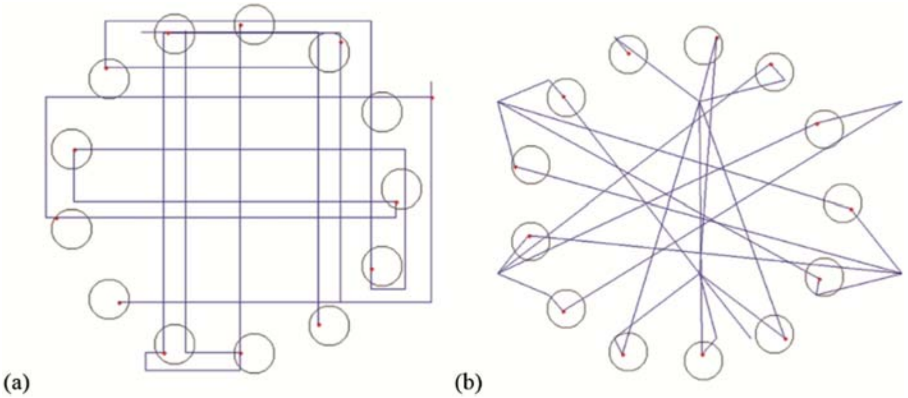

Fig. 6. Pointer trace examples in the case study for (a) the CKM input method and (b) the DualMouse input method. For both examples, $A = 500$ pixels and $W = 60$ pixels

We noted above that, as a percent, the performance advantage in throughput for CKM over DualMouse was similar (≈ 60 %) in the user study and the case study. The same percent difference is apparent when contrasting the results between the user study and the case study. See Table 1. The shaded cells contrast the results "by study" for movement time (s), error rates (%) and throughput (bits/s). The throughput for the participant in the case study was about 60 % lower than the mean throughput for the participants in the user study. This is true both for the CKM input method and the DualMouse input method. This result reflects an overall performance disadvantage for the user in the target community compared to non-disabled users.

Table 1. Comparison of results for user study and case study. Shaded cells show the difference (%) between studies

Input Method	Study	Movement Time (s)		Error Rate (%)		Throughput (bits/s)	
		Mean	diff	Mean	diff	Mean	diff
CKM	User study (n = 8)	6.9		2.42		0.52	
	Case study (n = 1)	7.83	+13.5%	5.77	+138.4%	0.21	-60.2%
DualMouse	User study (n = 8)	12.1		2.88		0.31	
	Case study (n = 1)	11.78	-2.7%	1.92	-33.2%	0.13	-59.0%

For movement time (s), the user study and case study results were similar. For error rate (%), the case study participant had a lower error rate with the DualMouse input method, but a substantially higher rate for the CKM input method.

6 Conclusion

The objective of this paper is not to introduce OSDS nor its mouse modes, as both have been introduced elsewhere. Rather, the contribution is in the evaluation. This is the first time the innovative stepwise mouse control method implemented in OSDS is compared to the well-known keyboard mouse technique involving a Fitts' law task.

The results of this comparison are not only relevant for the development process of OSDS (and thus prospective future users). In addition, the results have a direct impact on research on mouse-replacement interfaces as a whole. Potential beneficiaries are users who are unable to use a standard mouse.

Despite multiple promising attempts to evaluate the software involving a larger group of members of the target population (e.g., [12]), the developer of OSDS himself so far remains the only truly committed test subject. However, it is hoped that this research will assist in motivating new users. In short, the time to get familiar with the tool is ultimately repaid. The more successful users have already expressed positive experiences.

This paper together with the text entry study involving non-disabled participants presented two years ago [4] mark the start of a series of user tests. The overall goal is to evaluate the entirety of OSDS features, positioning it as a useful assistant for persons with certain physical disabilities.

Acknowledgments. This work is partially supported by DFG grant FE 936/6-2 "EFFENDI – EFficient and Fast text ENtry for persons with motor Disabilities of neuromuscular orIgin".

References

1. Branson, D., Demchak, M.: The use of augmentative and alternative communication methods with infants and toddlers with disabilities: a research review. Augmentative Altern. Commun. **25**(4), 274–286 (2009)
2. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale (1983)
3. Felzer, T., MacKenzie, I., Rinderknecht, S.: DualScribe: a keyboard replacement for those with Friedreich's ataxia and related diseases. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) ICCHP 2012, Part II. LNCS, vol. 7383, pp. 431–438. Springer, Heidelberg (2012)
4. Felzer, T., MacKenzie, I., Rinderknecht, S.: Applying small-keyboard computer control to the real world. In: Miesenberger, K., Fels, D., Archambault, D., Peňáz, P., Zagler, W. (eds.) ICCHP 2014, Part II. LNCS, vol. 8548, pp. 180–187. Springer, Heidelberg (2014)
5. Felzer, T., MacKenzie, I.S., Rinderknecht, S.: Efficient computer operation for users with a neuromuscular disease with OnScreenDualScribe. J. Interact. Sci. **2**(2) (2014)

6. Felzer, T., Rinderknecht, S.: Mouse mode of OnScreenDualScribe: three types of keyboard-driven mouse replacement. In: CHI EA, pp. 1593–1598. ACM (2013)
7. Jansen, A., Findlater, L., Wobbrock, J.O.: From the lab to the world: lessons from extending a pointing technique for real-world use. In: CHI EA,. pp. 1867–1872. ACM (2011)
8. John, B.E., Kieras, D.E.: Using GOMS for user interface design and evaluation: Which technique? ACM Trans. Comput.-Hum. Interact. (TOCHI) **3**(4), 287–319 (1996)
9. MacKenzie, I.S.: Human-Computer Interaction: An Empirical Research Perspective. Elsevier, Amsterdam (2013)
10. Magee, J., Felzer, T., MacKenzie, I.S.: Camera Mouse + ClickerAID: Dwell vs. single-muscle click actuation in mouse-replacement interfaces. In: Antona, M., Stephanidis, C. (eds.) Universal Access in HCI. LNCS, vol. 9175, pp. 74–84. Springer, Heidelberg (2011)
11. Newell, A.F., Gregor, P., Alm, N.: HCI for older and disabled people in the Queen Mother Research Centre at Dundee University, Scotland. In: CHI EA, pp. 299–302. ACM (2006)
12. Saulynas, S., Albar, L., Kuber, R., Felzer, T.: Using OnScreenDualScribe to support text entry and targeting among individuals with physical disabilities. In: Proceedings of ASSETS, pp. 303–304. ACM (2015)
13. Soni, D.M.: A review on Friedreich Ataxia, a neurodegenerative disorder. J. Drug Deliv. Ther. **4**(6), 45–52 (2014)
14. Soukoreff, R.W., MacKenzie, I.S.: Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. Int. J. Hum.-Comput. Stud. **61**(6), 751–789 (2004)
15. Williams, M.R., Kirsch, R.F.: Evaluation of head orientation and neck muscle EMG signals as command inputs to a human-computer interface for individuals with high tetraplegia. IEEE Trans. Neural Syst. Rehabil. Eng. **16**(5), 485–496 (2008)
16. Wobbrock, J.O., Fogarty, J., Liu, S., Kimuro, S., Harada, S.: The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation. In: CHI, pp. 1401–1410. ACM (2009)
17. Zhai, S., Morimoto, C., Ihde, S.: Manual and gaze input cascaded (MAGIC) pointing. In: Proceedings of the CHI, pp. 246–253. ACM (1999)