Multi-Switch Scanning Keyboards: A Theoretical Study of Simultaneous Parallel Scans with QWERTY Layout

Frode Eika Sandnes[1,2(✉)], Evelyn Eika[1] and Fausto Orsi Medola[3]

[1]Department of Computer Science, Faculty of Technology, Art and Design, OsloMet – Oslo Metropolitan University, Oslo, Norway
[2]Faculty of Technology, Westerdals Oslo School of Art, Communication and Technology, Oslo, Norway
[3]São Paulo State University (UNESP), School of Architecture, Arts and Communication, Bauru, Brazil.
frodes@hioa.no, Evelyn.Eika@hioa.no, fausto.medola@faac.unesp.br

Abstract. Scanning keyboards can be useful aids for individuals with reduced motor function. However, scanning input techniques are known for being very slow to use because they require waiting for the right cell to be highlighted during each character input cycle. This study explores the idea of parallel scanning keyboards controlled with multiple switches and their theoretical effects on performance. The designs explored assume that the keyboard layouts are familiar to users and that the mapping between the switches and the keyboards are natural and direct. The results show that the theoretical performance increases linearly with the number of switches used. Future work should perform user tests with parallel scans to assess the practicality of this approach.

Keywords: scanning keyboards, text entry, joystick, parallel scans, reduced motor function.

1 Introduction

Individuals with reduced motor function may be unable to use a regular keyboard. It may be due to tremor, stiffness, missing hand or arm, or the total inability to use arms or hands. One solution is to control computers with a switch, either controlled with the hand, the head, gaze, eye blink, or a brain-computer interface. Much research effort has gone into the use of single switches to perform text entry by the means of scanning keyboards [1].

Scanning keyboards are time-consuming to use since most of the time is involved in waiting for the target to be highlighted. Therefore, most studies into scanning keyboards have focused on layout optimizations [2, 3, 4], word prediction [5], ambiguous keyboards [6], and reducing errors [7, 8].

This study explores the concept of using several switches in parallel to improve text entry performance. By using several switches, multiple scanning sequences can be performed in parallel. One can thereby possibly achieve performance improvements.

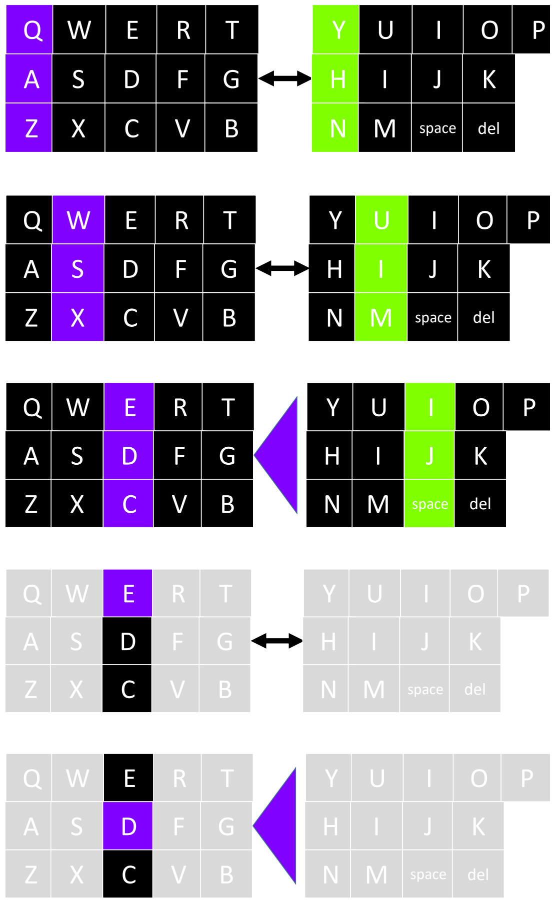

Fig. 1. Two-level scanning keyboards with double switch (column-cell).

2 Background

The literature on scanning keyboards is vast; see, for instance, the survey by Polacek [1]. A typical scanning keyboard is usually implemented by displaying a virtual keyboard where the rows are displayed in a sequential order. Once the row with the desired character is displayed, it can be selected by activating the switch. Next, the cells of the selected row are shown one-by-one for a certain dwell-time. Once the desired cell is highlighted, it is selected by activating the switch. Research into scanning keyboards has attempted to reorganize the virtual keyboards into more efficient layouts with shorter distances to the more frequent characters [2, 3, 4]. Word prediction is also commonly used to help users speed up the text entry [5]. Keyboards with multiple characters assigned to each cell combined with dictionaries to resolve ambiguities were also attempted [6].

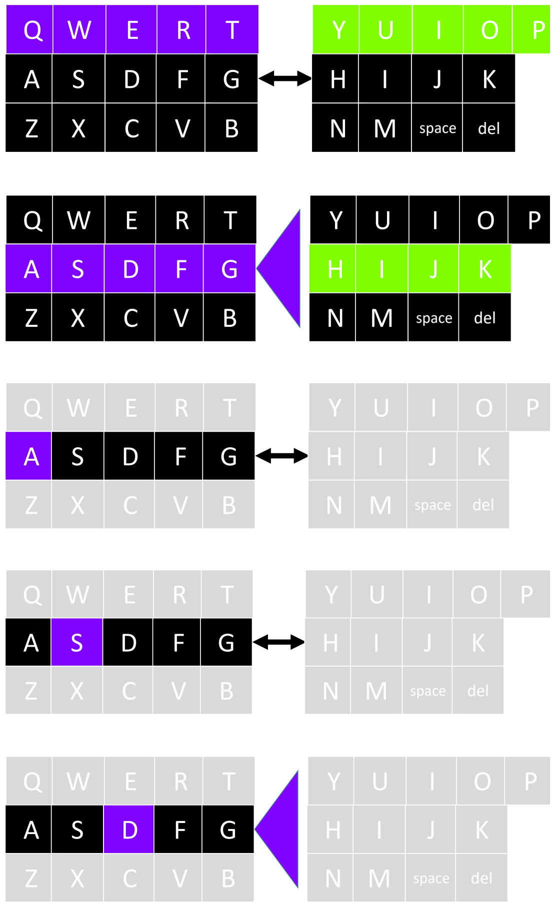

Fig. 2. Two-level scanning keyboard with double switch (row-cell).

A number of studies have also explored error patterns that are specific to scanning-based text entry [7, 8, 9].

Alternatives to scanning keyboards include Morse code [10], which can be input using a single switch with long and short presses and tapping [11, 12] by the means of two-dimensional tapping codes. Another different area of research includes chording [13, 14] where a finger is only assigned one key, but several keys are often pressed simultaneously. Chording also allows for certain error correction mechanisms [15, 16]. Other approaches include menu-based systems [17, 18] where the user finds the desired character by going through some menu structure, gestures [19, 20], prefix-based prediction [21], and abbreviations expansion [22] where the user could just enter key consonants to save effort and these abbreviations are expanded using dictionaries. In addition to simplifying text entry, text prediction [21] and other query building aids have also shown to be beneficial for users with dyslexia [23, 24]. Indirect information input based on proximity is another totally different approach suitable for specialized applications [25, 26].

It has been pointed out that scanning keyboard research is challenging due to the difficulties of recruiting participants from the target group and several researchers have therefore proposed various performance models [9, 27, 28]. Of interest herein are the scan steps per character [28] which is the sum of the number of scan steps to reach a character multiplied by the frequency of that particular character.

3 Input Methods

This section describes the multi-switch input techniques explored in this study. Common to all the techniques is that they are based on the QWERTY layout since users are more likely to accept new input techniques if they can reuse existing skills and are exposed to familiar elements [29, 30]. One important principle is that there should be a direct mapping between the controls and the virtual keyboard such that learning time is minimized.

3.1 Double Switch

A simple enhancement of a simple switch is a double switch, which is a spring-loaded device that can be moved left or right and that will return to its original position once it is released. It would typically be operated by an individual with very limited motor function who can perform very coarse-grained motions. A double switch can, for instance, be implemented using a single joystick where all leftwards motions are interpreted as West and all rightwards motions interpreted as East.

Unlike the other designs described herein, the double switch requires characters to be selected in two steps: first the group, then the cell. Two designs are explored : column-cell and row-cell. The Column-cell-design first scans the five columns on each side of the keyboard in parallel during the first step. The user then selects the left or right side when the desired column is determined. During the second step the three cells within the selected column is selected by any motion with the joystick. This gives the user freedom in the direction of the second motion allowing the user to make a more natural continuous motion, that is, left-to-right, instead of left-release-left.

With the second variation, the user first selects the desired row on the left or right side when it is highlighted. During the second step the user selects the desired cell within the group with either a left or right motion with the switch. Fig. 1 illustrates the character grouping and switch mapping using the row-cell and Fig. 2 column-cell strategies. In both cases, it is simulated that the user wants to input the character D. With the column-cell strategy, the five columns for each hand are scanned in parallel. Once the third column on each side is reached, the user moves the joystick left to select the left column (EDC) and not the right column (I, J, SPACE). Next, the three cells EDC of the column are scanned and the user selects the second one (D) once it is highlighted by moving the joystick in any direction. In total, the user needs to make two decisions and two actions and wait for five scan steps.

The procedure is similar with the row-cell strategy, with the difference that the entire rows are scanned first. Once the second row is displayed, the left part of the row (ASDFG) is selected by moving the joystick left.

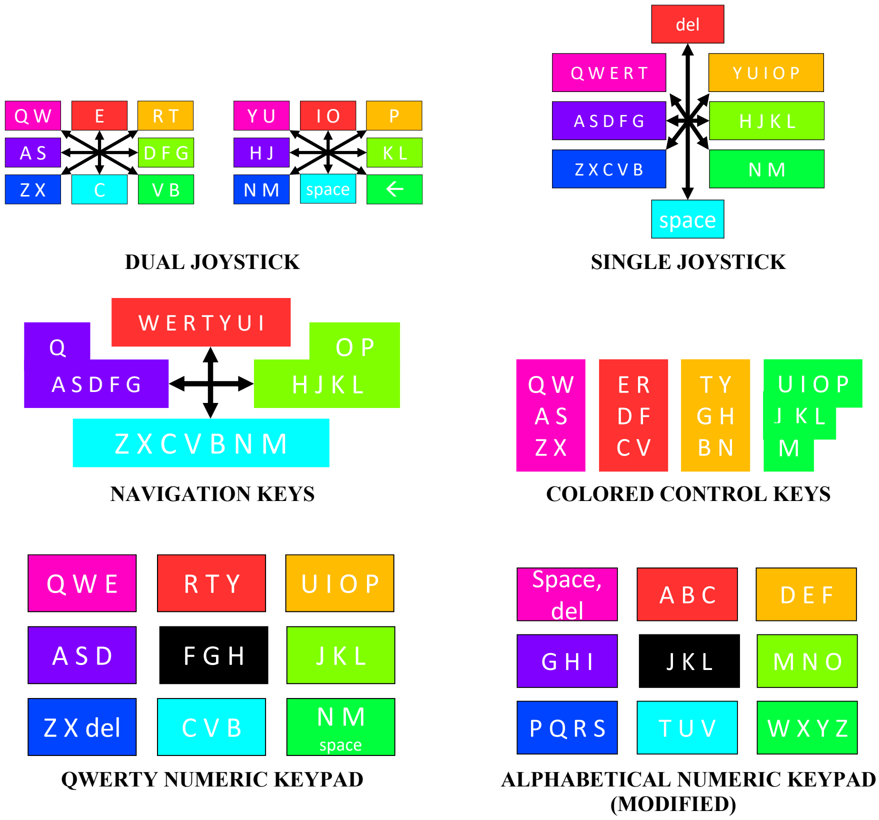

Fig. 3. Multi-switch input methods: a) dual joystick, b) single joystick, c) navigation keys, d) four colored control keys, e) QWERTY numeric keypad, and f) alphabetical numeric keypad. Colors are used to highlight direction.

Next, the cells of the row ASDFG are scanned and the user selects the third cell D once it is highlighted. Again, the user needs to make two decisions and take two actions and have to wait five scanning steps in total.

3.2 Double Joystick

The double joystick, or game controller, comprises two joysticks where each joystick can be moved in one of eight directions, or 16 directions for the two joysticks. Previously, text input for joysticks has been proposed by the means of an ambiguous keyboard where users would imagine each joystick being placed in the middle of each side of the keyboard and the joystick moved in the direction of the desired characters [29].

The double joystick input technique proposed herein comprises dividing the keyboard into two halves and with the joysticks virtually located between the S and D keys and between the J and K keys (see Fig. 3a). For each hand, a selection is made by moving the joystick in the direction of the desired group once the desired character of that group is highlighted.

An advantage of this input strategy is that each group is relatively small comprising 1 to 3 characters. Therefore, the penalty of missing a character during a scan is not too large.

Obviously, a double joystick requires the operation of both hands, and may therefore not be usable for people who are unable to use the hands in this way. However, in situations that allow for bimanual input, the double joystick may host the possibility of more efficient text input because of the 16 unique directions.

3.3 Single Joystick

Single joystick text input may be used for individuals who perhaps may only use one hand or may use some alternative input device such as a mouth-controlled switch. The most noteworthy research into single joystick text entry is by the means of gestures. The proposed approach requires much simpler single strokes compared to the more complex motions required to input gestures.

The single joystick allows the user to select one of eight directions. The QWERTY keyboard was therefore divided into six groups, comprising a left and a right part with three rows in each. Each of these six groups was accessed by North-West, West and South-West and North-East, East and South-East. Most of the groups are thus assigned five cells or characters. Space is assigned South and backspace is assigned North. Fig. 3b illustrates the single joystick character grouping and mapping.

3.4 Navigation Keys

The navigation keys design could be applicable to individuals who are unable to accurately control a single joystick in eight directions but can control the four easiest directions North, East, South, and West. Alternatively, it could be non-disabled users who need to input text with a limited input device such as a Smart TV remote control or an in-flight entertainment system controller.

The navigation maps directly to North, East, South, and West. Clearly, the QWERTY keyboard does not naturally partition into four groups according to such a division. However, an attempt is illustrated in Fig. 3c. The idea behind the partitioning is that the origin is the center of the keyboard located between the G and H buttons and the characters are divided into characters above, below, to the left, and to the right. Moreover, the group sizes were balanced such that no groups have more than seven cells or less than six cells.

3.5 Colored Control Buttons

Four colored control keys can be found on SmartTV remote controls and on some in-flight entertainment systems. Clearly, entertainment for individuals with limited input device characteristics is the intended application area.

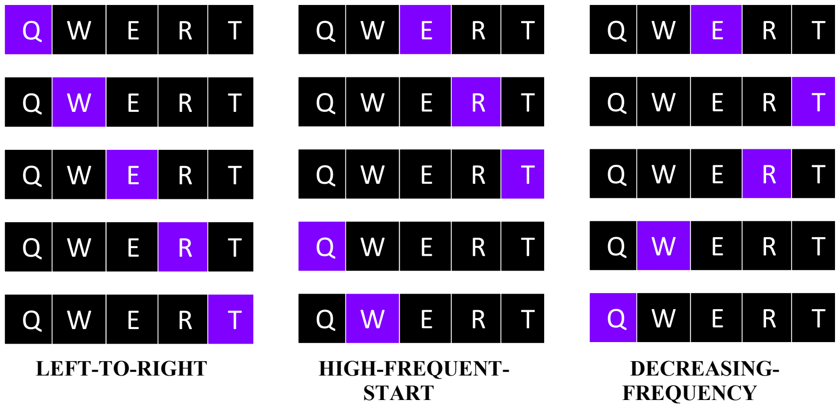

Fig. 4. Scanning sequences

With this design, the keyboard is partitioned into four groups along vertical lines of division comprising three groups of six cells and one group with eight cells at the right. The design is illustrated in Fig. 3d.

3.6 Numeric Keypad

Numeric keypads as text input devices have been explored in the text entry literature in terms of mobile text entry. The main strategies include multi-tap and ambiguous text entry. In this study, a parallel scan pattern is assigned to each key of the numeric keypad, where the desired letter is selected by pressing the assigned keypad button once the character is highlighted.

Two keypad assignments are made, namely to evenly distribute the characters across all the keys and to reuse the familiar numeric keypad character assignments as found on mobile phone keypads. The two key assignments are illustrated in Fig. 3e and 3f, respectively.

3.7 Scanning Sequence

For each of the designs described in the previous sections, three variations are explored: left-to-right, highest-frequency-letter-first, and decreasing-frequency. The left-to-right scanning sequence starts with the left-most character and goes towards the right. When the rightmost character is reached, the scan starts over again at the left. In cultures with left-to-right writing systems, it is believed that there is a preferential bias to go from left to right [31]. In the example in Fig. 4 (left) the scan goes via the sequence Q, W, E, R, T and start over with Q.

The second variation also goes from left to right, but starts with the most frequent character. This is illustrated in Fig. 4 (middle) where the scan starts with the most frequent character E and continues with R, T, Q, W.

The third variation involves scanning the character in decreasing order of frequency. For the sequence in Fig. 4 (right) this is E, T , R, W and Q.

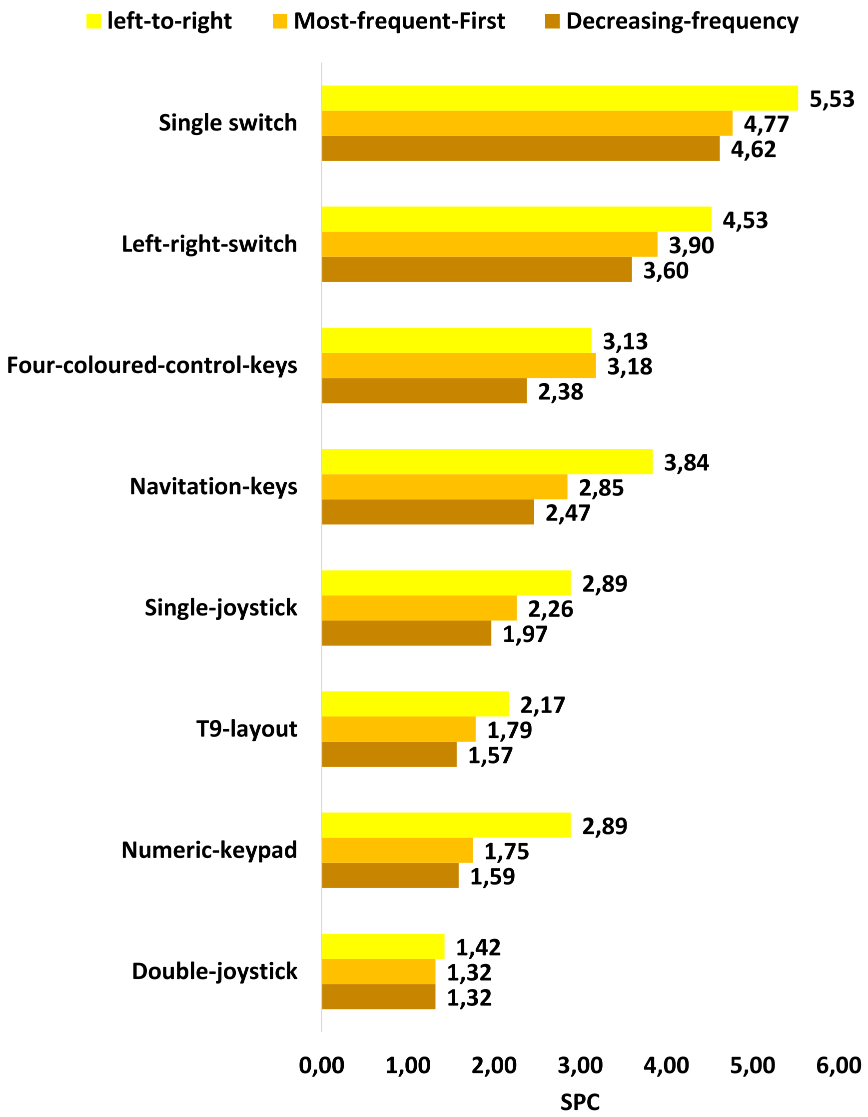

Fig. 5. SPC for the scanning keyboard designs with left-to-right scanning sequences (yellow), left-to-right-starting with the highest frequency letter scanning sequence (orange) and decreasing order of frequency (brown)

4 Results

Fig. 5 shows the effort required for each of the designs in terms of scanning steps per character [28]. A single-switch input strategy is provided as a baseline [32]. Each of the strategies is evaluated under three scanning sequence conditions, namely: from left-to-right, from-left-to-right but starting with the highest frequency character in the group, and scanning sequence according to decreasing frequency of occurrence (see Fig. 4). For each instance, it is assumed that the scanning pattern is repeated if the user does not act during a scan.

As expected, the single key strategy requires the most scan steps per character (5.53), while the double joystick requires the fewest (1.42 SPC). By replacing the single switch with a double switch, the saving is about one scan step per character. Further, the single joystick reduces the number of scan steps by 1.5 in comparison to the double switch, and it is nearly half the number of scan steps required with a single switch. The single joystick is obviously more complex to control, requiring more accurate motor control than the double switch. The number of scan steps with the dual joystick is only a small fraction of what is required with the single switch. However, the dual joystick required bimanual operation and quite detailed motor operations.

Several keyboard-based input strategies were included. These are not necessarily optimal for individuals with reduced motor function but may be suitable for individuals with limited input hardware such as text entry via remote controls or simple entertainment system controllers. The results show that the four-button design (3.13 SPC) performs better than the navigation key design (3.84 SPC). The reason for this is probably partially chance, that is, it happens that the division of the keyboard is more beneficial with the four colored keys than the navigation keys. Moreover, four navigation directions (north, east, south, and west) do not naturally map onto the QWERTY layout. However, the trend is reversed when scanning from left-to-right starting with the most frequent character, and the two techniques are nearly similar when scanning the characters in decreasing order of frequency.

Two numeric keypads are also included for reference, one based on the QWERTY layout (nonstandard mapping) and the other an alphabetical mapping (standard mapping). The results show that both numeric keypad-based techniques give fewer scan steps per character compared to the other techniques based on fewer switches. Note that the QWERTY based mapping uses nine switches while the alphabetical (standard) mapping only uses eight as the 1-key is not used. Surprisingly, the alphabetical layout yields a lower number of scan steps per character when each group is scanned from left-to-right.

Although the numeric keypad yields fewer scan steps per character, it may be more challenging to use compared to control keys or especially navigation keys. This is because the user probably will have to perform a visual search for the right key before pressing the key. With navigation keys the mapping may be more intuitive.

Another interesting observation is that the alphabetical numeric keypad only uses eight switches for characters. Yet, its performance (2.17 SPC) is much better than the QWERTY-based method for single joysticks (2.89), which also have eight switches. However, the single joystick-based method uses the familiar QWERTY layout and two of the directions are assigned to backspace and space. Simply, by assigning the characters of the alphabetical numeric keypad to the eight joystick directions the scan steps per character could be reduced quite significantly. However, this assignment would come at the expense of being unfamiliar and more complex access to the space and backspace characters.

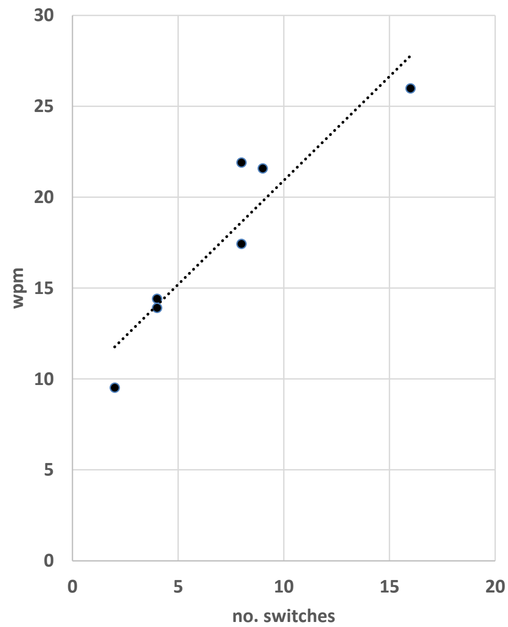

Fig. 6. Theoretic upper bound of wpm as a function of number of switches.

To explore the effects of switch quantity, the results obtained with the techniques discussed herein were visualized. Fig. 6 shows a scatterplot of the number of switches plotted against a theoretical measure of words per minute. The plot shows that the techniques explored approximately follow a line. However, it is important to note that the measures of words per minute are theoretical and not based on actual measurements.

An important difference between the double switch technique and the other techniques discussed herein is that the former requires two scanning levels, while the latter only requires one scanning level. One may expect that it is more tolerable to use a scanning keyboard with just one scanning level as it should be cognitively less demanding requiring only one action per character.

5 Conclusions

This study explored the effects of using multiple switches with parallel scanning keyboards based on a theoretical model. The results show that that it is theoretically possible to achieve an approximate linear increase in text entry rates with the number of switches. This study has explored this phenomenon under the assumptions that the letters must have a familiar layout (QWERTY) and a natural and direct mapping. Several configurations were explored, opening up for the possibility to tailor a particular design according to the characteristics of a user's motor abilities. It is also possible that scanning keyboards may be beneficial for non-disabled users in resource constrained systems that only allowprimitive input devices such as entertainment systems. This study is a theoretical back-of-the envelope evaluation of several text entry designs. Theoretical evaluations are by no means a valid substitute for performing measurements with real users, however, theoretical models can sometimes help identify interesting designs worth further study [33]. Future work will explore how users respond to parallel scanning keyboard designs.

References

1. Polacek, O., Sporka, A. J., Slavik, P.: Text input for motor-impaired people. Universal Access in the Information Society 16, 51-72 (2017)
2. Abascal, J., Gardeazabal, L., Garay, N.: Optimisation of the selection set features for scanning text input. In: International Conference on Computers for Handicapped Persons, pp. 788-795. Springer Berlin Heidelberg (2004)
3. Higger, M., Moghadamfalahi, M., Quivira, F., Erdogmus, D.: Fast Switch Scanning Keyboards: Minimal Expected Query Decision Trees. arXiv preprint arXiv:1606.02552 (2016)
4. Zhang, X. C., Fang, K., Francis, G.: Optimization of switch keyboards. In: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility. ACM (2013)
5. Jones, P. E.: Virtual keyboard with scanning and augmented by prediction. In: Proceedings of the 2nd European Conference on Disability, Virtual Reality and Associated Technologies, pp. 45-51 (1998)
6. Mackenzie, I. S., Felzer, T.: SAK: Scanning ambiguous keyboard for efficient one-key text entry. ACM Transactions on Computer-Human Interaction (TOCHI) 17, (2010)
7. Zhang, X., Fang, K., Francis, G.: How to optimize switch virtual keyboards to trade off speed and accuracy. Cognitive Research: Principles and Implications 1, 6 (2016)
8. Simpson, R. C., Mankowski, R., Koester, H. H.: Modeling one-switch row-column scanning with errors and error correction methods. The Open Rehabilitation Journal 4, 1-12 (2011)
9. Francis, G., Johnson, E.: Speed–accuracy tradeoffs in specialized keyboards. International Journal of Human-Computer Studies 69, 526-538 (2011)
10. Levine, S., Gauger, J., Bowers, L., Khan, K.: A comparison of Mouthstick and Morse code text inputs. Augmentative and Alternative Communication 2, 51-55 (1986)
11. Sandnes, F. E., Medola, F. O.: Exploring Russian Tap-Code Text Entry Adaptions for Users with Reduced Target Hitting Accuracy. In Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, pp. 33-38. ACM (2016)
12. Gong, J., Haggerty, B., Tarasewich, P.: An enhanced multitap text entry method with predictive next-letter highlighting. In :CHI'05 extended abstracts on Human factors in computing systems, (pp. 1399-1402. ACM (2005)
13. Sandnes, F.E.: Can spatial mnemonics accelerate the learning of text input chords? In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 245–249. ACM (2006)
14. Sandnes, F. E.: Human performance characteristics of three-finger chord sequences. Procedia Manufacturing 3, 4228-4235 (2015)
15. Sandnes, F. E., Huang, Y. P.: Chording with spatial mnemonics: automatic error correction for eyes-free text entry. Journal of information science and engineering 22, 1015-1031 (2006)
16. Sandnes, F. E., Huang, Y. P.: Chord level error correction for portable Braille devices. Electronics Letters 42, 82-83 (2006)

17. Sandnes, F.E., Thorkildssen, H.W., Arvei, A., Buverad, J.O.: Techniques for fast and easy mobile text-entry with three-keys. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences. IEEE (2004)
18. Sandnes, F.E., Jian, H.L., Pair-wise variability index: Evaluating the cognitive difficulty of using mobile text entry systems. In: International Conference on Mobile Human-Computer Interaction. LNCS, vol. 3160, pp. 347-350. Springer Berlin Heidelberg (2004)
19. Sandnes, F.E., Tan, T.B., Johansen, A., Sulic, E., Vesterhus, E., Iversen, E.R.: Making touch-based kiosks accessible to blind users through simple gestures. Universal Access in the Information Society 11, 421-431 (2012)
20. Perlin, K.: Quikwriting: continuous stylus-based text entry. In: Proceedings of the 11th annual ACM symposium on User interface software and technology, pp. 215-216. ACM (1998)
21. Darragh, J. J., Witten, I. H., James, M. L.: The reactive keyboard: A predictive typing aid. Computer 23, 41-49 (1990)
22. Sandnes, F. E.: Reflective text entry: a simple low effort predictive input method based on flexible abbreviations. Procedia Computer Science 67, 105-112 (2015)
23. Berget, G., Sandnes, F.E.: Do autocomplete functions reduce the impact of dyslexia on information searching behaviour? A case of Google. J. Am. Soc. Inf. Sci. Technol. 67, 2320–2328 (2016)
24. Berget, G., Sandnes, F.E.: Searching databases without query-building aids: implications for dyslexic users. Inf. Res. 20 (2015)
25. Huang, Y. P., Chang, Y. T., Sandnes, F. E.: Ubiquitous information transfer across different platforms by QR codes. Journal of Mobile Multimedia 6, 3-13 (2010)
26. Sandnes, F. E., Herstad, J., Stangeland, A. M., Orsi Medola, F.: UbiWheel: A Simple Context-Aware Universal Control Concept for Smart Home Appliances that encourages Active Living. In: Proceedings of Smartworld 2017, pp. 446-451. IEEE (2017)
27. Bhattacharya, S., Samanta, D., Basu, A.: Performance models for automatic evaluation of virtual scanning keyboards. IEEE Transactions on neural systems and rehabilitation engineering 16, 510-519 (2008)
28. MacKenzie, I. S.: Modeling text input for single-switch scanning. In: International Conference on Computers for Handicapped Persons, pp. 423-430. Springer Berlin Heidelberg. (2012)
29. Sandnes, F.E., Aubert, A.: Bimanual text entry using game controllers: relying on users' spatial familiarity with QWERTY. Interact. Comput. 19, 140–150 (2007)
30. Sandnes, F. E.: Effects of common keyboard layouts on physical effort: Implications for kiosks and Internet banking. In: Sandnes, F.E., Lunde, M. Tollefsen, M., Hauge, A.M., Øverby, E., Brynn, R. (eds.) The proceedings of Unitech2010: International Conference on Universal Technologies, pp. 91-100. Tapir Academic Publishers (2010)
31. Sandnes, F.E.: Directional bias in scrolling tasks: a study of users' scrolling behaviour using a mobile text-entry strategy. Behav. Inf. Technol. 27, 387–393 (2008)
32. Sandnes, F. E., Medola, F. O.: Effects of Optimizing the Scan-Path on Scanning Key-boards with QWERTY-Layout for English Text. Studies in health technology and informatics 242, 930-938 (2017)
33. Sandnes, F. E.: Evaluating mobile text entry strategies with finite state automata. In: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, pp. 115-121. ACM (2005)