

DualScribe: A Keyboard Replacement for Those with Friedreich's Ataxia and Related Diseases

Torsten Felzer¹, I. Scott MacKenzie², and Stephan Rinderknecht¹

¹ Institute for Mechatronic Systems, Technische Universität Darmstadt,
Darmstadt, Germany

² Department of Computer Science and Engineering, York University,
Toronto, Canada M3J 1P3

{felzer,rinderknecht}@ims.tu-darmstadt.de, mack@cse.yorku.ca

Abstract. An alternative text composition method is introduced, comprising a small special-purpose keyboard as an input device and software to make text entry fast and easy. The work was inspired by an FA (Friedreich's Ataxia) patient who asked us to develop a viable computer interaction solution – taking into account the specific symptoms induced by his disease. The outcome makes text entry easier than with the standard keyboard without being slower. It is likely that the system has general use for anyone with a similar condition, and also for able-bodied users looking for a small-size keyboard. We present a usability study with four participants showing the method's effectiveness.

Keywords: Human-computer interaction, special-purpose keyboard, word prediction, ambiguous keyboards, neuromuscular diseases, Friedreich's Ataxia.

1 Introduction

John is 41 years old and was diagnosed 20 years ago with the inherited neuromuscular disease Friedreich's Ataxia (FA). He has considerable motor problems, worsened by secondary neurological symptoms affecting his sight and speech. He depends on a computer for everyday work, in particular to compose text documents. Lacking viable alternatives, John still uses a standard manual keyboard for text entry. However, this requires substantial time and effort. The entry rate is very low, sometimes only 2–3 wpm, and is decreasing as the disease progresses.

Our work sought to make John's life easier by developing an efficient, effortless, and practical alternative to the standard keyboard. The alternative is tailored to his needs which are a direct consequence of his disease. The resulting solution fully utilizes what someone with FA can do, while compensating for what such a person cannot do. The system is therefore not only usable by John, but by anyone with similar conditions.

The work is significant since the estimated prevalence of inherited neuromuscular disorders is one person in 3,000 [1]. In the USA, for example, that translates

into more than 100,000 people. The work is also important to text entry in general. In addition to use in the workplace, using a computer for e-mail, Internet chat, texting, etc., is unthinkable without a quick and efficient means to enter text.

In the next section, we examine common alternatives for persons with disabilities and evaluate them in view of someone with a neuromuscular disease. After that, we describe the newest version of our text composition application: *DualScribe*, which is tailored to the needs of that particular user group. This is followed by the description of a usability study with four participants, including John and three able-bodied computer users. The paper is concluded with a short summary and some thoughts on future work.

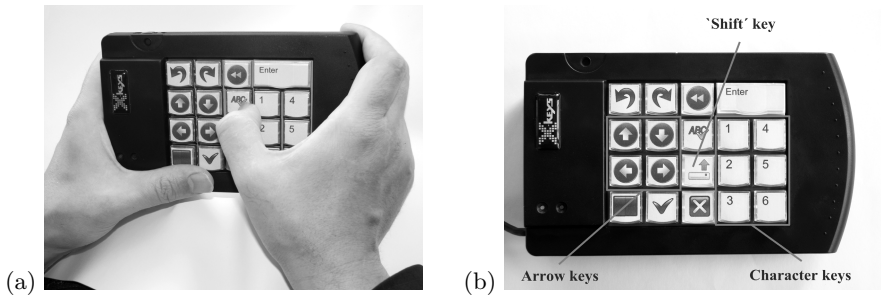


Fig. 1. *DualPad*: Special-purpose keyboard used as replacement for the standard keyboard. It consists of four rows and five columns and has 18 active keys (not $4 \times 5 = 20$), because the rightmost key in row 1 is a double-width key and the leftmost key in row 4 is deactivated. Columns 1 and 2 are operated with the left thumb, while the right thumb is responsible for columns 3-5. (a) With hands. (b) Without hands.

2 Literature Review

There are many ways to assist persons with motor disabilities to enter text. These may be categorized by looking at the targeted input device. First, there are standard techniques, such as word prediction, which aims to save keystrokes. After each character, a list of extensions of the string is presented, and the user can select among those candidates. This reduces the physical load on the user, but looking through the list to find the desired word increases the cognitive load; so, word completion is not always faster [2]. Since someone with FA usually has motor problems leading to unsteady movement, utilizing a full-size keyboard may be problematic due to key-to-key “travel” distances.

Dasher [3] is an assistive tool for a variety of pointing devices, in the simplest form a standard mouse. It allows the user to enter text quickly (normally!) by moving the mouse pointer into areas on the screen associated with characters. Because of missing fine motor control and reduced hand-eye coordination, this tool is not a suitable alternative for FA patients.

The second group of text entry methods focuses on switch-like devices. Input relies on a small number of signals with the help of a scanning scheme [4]. Instead of using a full keyboard, the user confirms suggestions made by the computer, for example, by actuating a key or switch. One example is a row-column scanning keyboard where the computer suggests first the row, and then the column within the row, of a character in a two-dimensional matrix. These systems (e.g., [5,6]) typically do not make full use of the physical capabilities of an FA patient (who can employ both hands, for example). So, input is unnecessarily cumbersome, resulting in a very low input rate. Editing tasks, such as rearranging a sentence, are particularly difficult.

The third approach involves contact-free devices. For those who cannot use their hands or arms, automatic speech recognition (ASR) is an efficient alternative (e.g., [7]). As the voice of an FA patient is usually subject to dysarthria, the recognition algorithm may falter due to speech variations. Non-verbal voice interaction (NVVI) – involving humming or whistling – might be an answer (e.g., [8]), but the vocal problems are often too overwhelming.

Another method for hands-free text entry is eye tracking, where the user's eyes can move a mouse cursor, for example, to select characters displayed on an on-screen keyboard (e.g., [9]). Since FA is often accompanied by pathological nystagmus, eye tracking is often impossible. An idea quite similar to eye tracking is the camera mouse [10], which tracks a freely selectable spot on the user's face with the help of a standard webcam. However, that may also be too difficult if the user's head movements are unsteady or uncontrollable.

Hamidi et al. [11] present an interesting text entry approach. It is also based on speech recognition, but in contrast to allowing arbitrary words, the system merely discriminates between a limited number of user-definable keywords which are translated into emulated keystrokes. In other words, instead of typing on a standard manual keyboard, the user can define, for example, "Marc" as "M" or "Tom" as "T", and the system only has to distinguish the keywords. This could be the starting point for an adequate contact-free alternative.

In our approach, we combine some of the ideas above and convert them to a text composition application called *DualScribe*, which is characterized by the special-purpose keyboard shown in fig. 1. This input device was chosen because of anecdotal evidence: John (who usually uses the middle fingers of both hands) likes to type the leftmost keys on the keyboard (like "A", "Q", "TAB", or left "SHIFT"), since he then can use his left thumb for typing while handling the left edge of the keyboard as a guide.

3 *DualScribe* – System Description

It is beyond the scope of this paper to detail every feature of *DualScribe*, so we focus only on the main concepts. In short, the software emulates virtual keystrokes in response to the physical keystrokes of the user on the 18 keys of the *DualPad*. The keys of the *DualPad* are not only associated with direct keystroke emulation, but also with changes in the program state of *DualScribe*.

The program offers a small number of modes with the mappings from keys to functionality depending on the currently active mode. Each key’s function is similar in every mode (although not always identical).

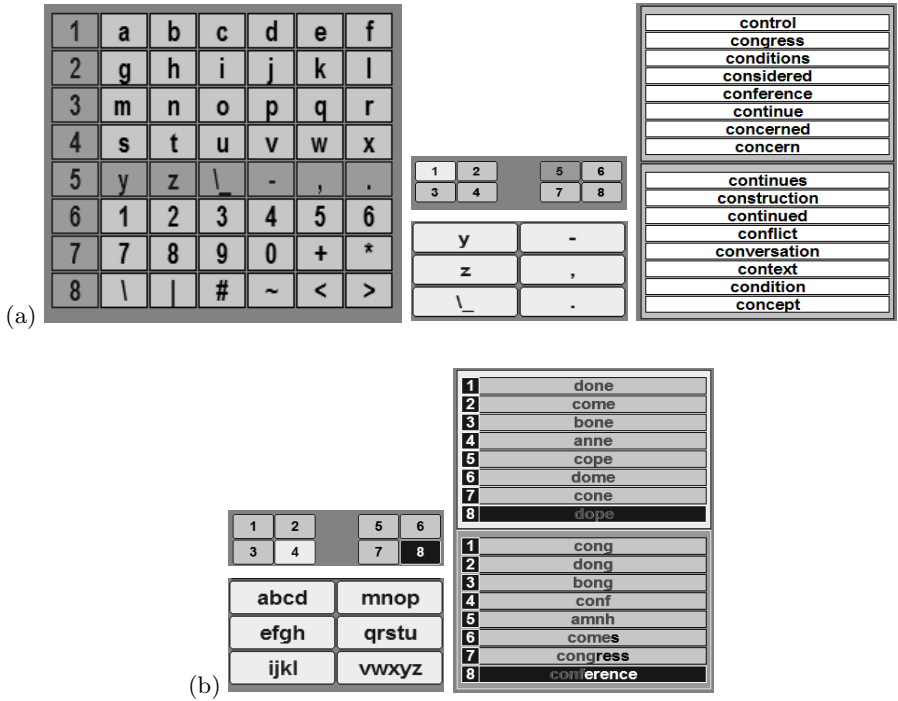


Fig. 2. Input ideas. (a) *Dual mode* with eight-by-six selection grid and a word prediction list after entering “con”, which requires pressing the keys “4”, then “3”, “←”, “3”, “←”, and finally “2” on the *DualPad*. (b) *Ambiguous mode* where each character key stands for four or five characters with candidate list after entering the sequence “1442”.

The default entry of a character (also called *dual mode*) involves two keystrokes (leading to the program’s name). The selectable characters are arranged in eight rows and six columns (as depicted in fig. 2a for lower-case letters). The user selects the row of a character with the help of the four arrow keys located on the *DualPad* and the column with the six character keys. In *single keystroke mode*, the arrow keys of the *DualPad* do the same as the arrow keys on the standard keyboard – that is the reason for their labeling. In *dual mode*, they could as well be labeled A, B, C, D, as they are simply four keys to directly select rows 1-4 – or 5-8 if pressed twice – of the 8 × 6 grid.

The *KSPC* (keystrokes per character) for the input method is 2.0; however, word prediction using a 100k-item dictionary (compiled from Ref. [12]) reduces *KSPC*. Considering all 100,000 words and always taking the shortest path to

enter a word [13] yields a *KSPC* of 1.1159. This is close to the inherent *KSPC* = 1.0 for a regular QWERTY keyboard.

There is also an *ambiguous mode* which is T9-like except only using six keys. As the user enters the characters “1”–“6”, *DualScribe* looks for matches in the same dictionary as is used for word prediction. A frequency-ordered list of up to 16 candidates (including extensions, but with priority given to *exact* matches) is suggested to the user (as two lists with eight entries each) who then selects the desired word in a manner similar to the row in *dual mode*. The implementation is analogous to MacKenzie and Felzer’s definition of a *scanning ambiguous keyboard* [14], except with direct selection instead of scanning.

Since the dictionary is large, almost every word (including some proper names) is known and can thus be entered without switching to *dual mode*. Moreover, the user can easily extend the dictionary, i.e., make arbitrary expressions known to the system. On the other hand, a large dictionary also implies a high ambiguity, which means that the candidate list may contain many matches and only few extended suggestions (as depicted in fig. 2b). In any event, this method leads to a low *KSPC* of 0.8807.

The user is not restricted to the 48 symbols shown in fig. 2a. Various mechanisms, such as a “SHIFT” key or a *menu mode*, let the user produce any character or symbol that is found on a standard keyboard. Besides, *DualScribe* has its own editor window and offers features that integrate with any regular text composition software, including a movable cursor, copy/paste, find/replace, undo/redo.

4 Empirical Data

A pilot study revealed that entering text in *ambiguous mode* is faster and more comfortable than in *dual mode* (which is not really surprising, given the lower *KSPC*). Therefore, *ambiguous mode* was chosen for a first field test using the German language version of the program. During an exhibition at the local university, visitors were given the opportunity to measure their text entry rate with *DualScribe* in comparison to the standard QWERTY keyboard.

The participants’ task was to transcribe – as fast and as accurately as possible – nine German phrases randomly drawn from a list of 56 phrases with lengths varying between 31 and 81 characters. All phrases contained upper- and lower-case letters and punctuation, and it was made sure that every single word was in the dictionary (based on the DeReWo [15] in the German version), so that all text could be entered in *ambiguous mode*.

The nine runs were divided into three blocks with three runs each. Participants were instructed to employ the *DualPad* in the first and third blocks and the standard keyboard in the second block. Correcting typing errors (e.g., by using backspace) was allowed, but not required, and the number of correction operations used in each run was noted.

In addition to John (for whom the system was originally developed), three able-bodied participants, one 26-year-old female participant, and two male participants (aged 45 and 27, respectively) completed the entire procedure. For the

study, the four participants – all frequent computer users – were coded P0 (for John), P1, P2, and P3. The resulting entry rates are depicted in fig. 3 for the *ambiguous mode* in *DualScribe* and in fig. 4 for the standard keyboard.

The most striking result is that John’s entry rate with *DualScribe* appears comparable to that of the able-bodied participants. Of course, the comparison is

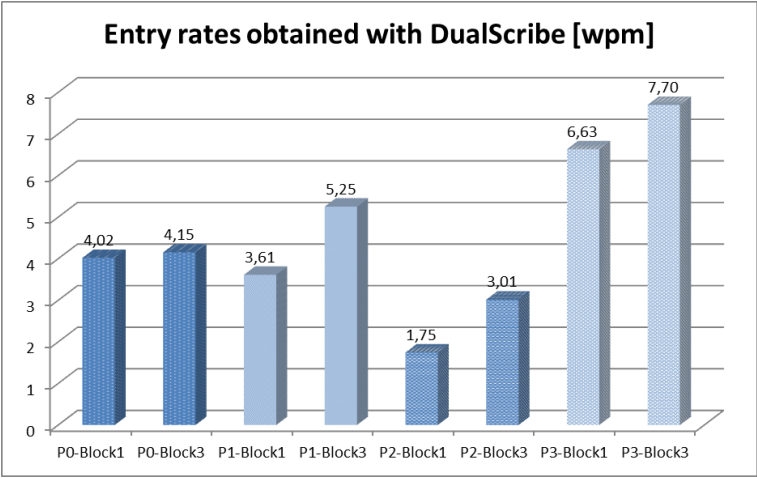


Fig. 3. Average text entry rates obtained in the four-participant usability study. Results for two blocks with three phrases each while the participants were using the *DualPad*.

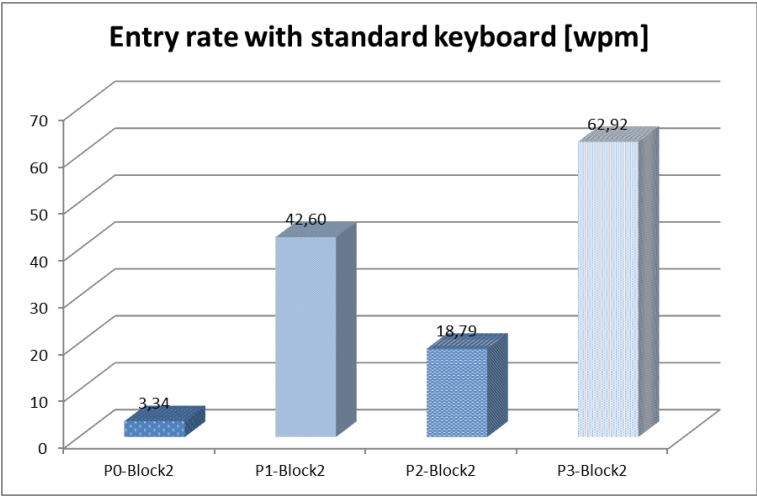


Fig. 4. Average text entry rates for one block while using the standard keyboard

a little unfair, since John was already familiar with the input method, while the visitors to the exhibition only practiced for a few minutes prior to the test (P1 for 2 minutes, P2 only for 1 minute, but P3 for at least 20 minutes). Anyway, his performance using *DualScribe* is not “that far off” as with the standard keyboard (with up to 20 times slower entry rate).

The practice time is also the main reason for the difference in blocks 1 and 3. John (P0) has a lot of experience with *DualScribe*, and a few more sentences can hardly improve his entry rate. However, this amount of practice is enough for the other participants to considerably improve their performance. Even though it seems likely that their learning curves will continue the upward trend, it has to be said that the built-in acceleration techniques cannot compensate for a full keyboard, if it can be operated fast. So *DualScribe* is a viable alternative for persons with certain disabilities, but for able-bodied users, it is only adequate in specific situations.

5 Conclusion

A text entry tool optimized for a small keypad which can replace a full-size keyboard has been introduced. The pad is held with both hands and operated with the thumbs only. The tool – an advancement of a simpler system involving a game controller [16] – is therefore ideal for someone with FA. A usability study justified that assumption. It was shown that it is possible for an FA patient to achieve text entry rates with the system that are somehow on par with those achieved by able-bodied computer users.

The fact that a first prototype is already in use currently (by a particular individual with FA, for composing private e-mails) also shows its usefulness. For the future, it is planned to extend the idea to computer operation in general (not just text composition). This requires directing virtual keystrokes to any active application (instead of a dedicated editor window), as well as adding a mouse mode.

Acknowledgments. This work is partially supported by DFG grant FE 936/6-1 “EFFENDI – EFFicient and Fast text ENtry for persons with motor Disabilities of neuromuscular origin”.

References

1. Lecky, B.R.F.: Neuromuscular disorders: Clinical and molecular genetics. *Brain* 122(4), 1–790 (1999)
2. Koester, H.H., Levine, S.P.: Modeling the speed of text entry with a word prediction interface. *IEEE Trans. Rehab. Eng.* 2(3), 177–187 (1994)
3. Ward, D.J., Blackwell, A.F., MacKay, D.J.C.: Dasher - a data entry interface using continuous gestures and language models. In: *Proc. UIST 2000*, pp. 129–137. ACM Press (2000)
4. Baljko, M., Tam, A.: Motor input assistance: Indirect text entry using one or two keys. In: *Proc. ASSETS 2006*, pp. 18–25. ACM Press (2006)

5. Wandmacher, T., Antoine, J.Y., Poirier, F., Départe, J.P.: Sibylle, an assistive communication system adapting to the context and its user. *ACM Trans. Access. Comput.* 1(1), 1–30 (2008)
6. Felzer, T., MacKenzie, I.S., Beckerle, P., Rinderknecht, S.: Qanti: A Software Tool for Quick Ambiguous Non-standard Text Input. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) *ICCHP 2010, Part II. LNCS*, vol. 6180, pp. 128–135. Springer, Heidelberg (2010)
7. Zhang, W., Duffy, V.G., Linn, R., Luximon, A.: Voice recognition based human-computer interface design. *Computers & Industrial Engineering* 37, 305–308 (1999)
8. Sporka, A.J., Felzer, T., Kurniawan, S.H., Poláček, O., Haiduk, P., MacKenzie, I.S.: Chanti: Predictive text entry using non-verbal vocal input. In: *Proc. CHI 2011*, pp. 2463–2472. ACM Press (2011)
9. Kumar, M., Paepcke, A., Winograd, T.: EyePoint: Practical pointing and selection using gaze and keyboard. In: *Proc. CHI 2007*, pp. 421–430. ACM Press (2007)
10. Gips, J., Betke, M., Fleming, P.: The camera mouse: Preliminary investigation of automated visual tracking for computer access. In: *Proc. RESNA 2000*, pp. 98–100. RESNA Press (2000)
11. Hamidi, F., Baljko, M., Livingston, N., Spalteholz, L.: CanSpeak: A Customizable Speech Interface for People with Dysarthric Speech. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) *ICCHP 2010, Part I. LNCS*, vol. 6179, pp. 605–612. Springer, Heidelberg (2010)
12. Davies, M.: Word frequency data from the Corpus of Contemporary American English (COCA), <http://www.wordfrequency.info> (January 27, 2011)
13. MacKenzie, I.S.: KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. In: Paternó, F. (ed.) *Mobile HCI 2002. LNCS*, vol. 2411, pp. 195–210. Springer, Heidelberg (2002)
14. MacKenzie, I.S., Felzer, T.: SAK: Scanning Ambiguous Keyboard for Efficient One-Key Text Entry. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17(3), 11:1–11:39 (2010)
15. Institut für Deutsche Sprache: Korpusbasierte Wortformenliste DeReWo, v-100000t-2009-04-30-0.1, mit Benutzerdokumentation. Programmbereich Korpuslinguistik, Mannheim, Deutschland (2009), <http://www.ids-mannheim.de/kl/derewo/>
16. Felzer, T., Rinderknecht, S.: Using a game controller for text entry to address abilities and disabilities specific to persons with neuromuscular diseases. In: *Proc. ASSETS 2011*, pp. 299–300. ACM Press (2011)