Single-Handed Eyes-Free Chord Typing: A Text-Entry Study

Adrian Tarniceriu, Pierre Dillenbourg, and Bixio Rimoldi

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
Email: adrian.tarniceriu@epfl.ch, pierre.dillenbourg@epfl.ch, bixio.rimoldi@epfl.ch

Abstract—**For most users, interacting with mobile computing devices requires visual commitment to the input mechanism. As a result, devices such as smartphones and PDAs are not suitable in situations when visual attention is already focused on another task. Chording devices do not have this drawback but require some training. We evaluate the performance of a key-to-character mapping for a 5-key chording device designed to minimize the learning phase. The subjects in our study were able to memorize the mapping in the first 45 minutes of training. After approximately 350 minutes, the average entry speed was 20 words per minute. The influence of having visual, audio or no feedback was also evaluated. We found that the typing rates were the same under all three conditions, but the error rates were the smallest in the absence of feedback (2.32%) and the largest when the users could see what they typed (3.41%).**

***Keywords*–chording keyboard, text entry, key mapping, feedback.**

I. Introduction

There are currently many methods for interacting with mobile computing devices; the most popular ones are graphical interfaces, voice command and text input. Graphical interfaces are probably the most user-friendly, but they require the user's visual attention, which makes them unsuitable in situations where vision is already committed, such as walking in a crowded place, pushing a shopping cart, riding a bike, and driving a car. Yet, it would be nice if we could send a quick note while walking to work, or if we could look up the characteristics and the location of an item while pushing a shopping cart. While exerting physical activities such as jogging or riding a bike, the smart phone could process biometric information and could respond to various queries. Even though more controversial, it would be nice if, while driving a car, we could issue commands like "find fastest way home" or "inform partner of arrival time". If we could issue such commands safely, there would be no problem for the on-board computer to estimate the arrival time and interact with the driver's smartphone to send a message to the partner. If we could issue such commands with voice, few people would be concerned about safety, but there are many problems associated with using voice to control a computer: the performance is not satisfactory, particularly in noisy places, there is an issue with privacy, and there are moments/places where other people would not appreciate hearing us speak aloud to our computer. Brain computer interfaces would solve all these problems, but the technology is not sufficiently developed yet. Although text is not the most user-friendly way to interact with a mobile device, it could be the most efficient in the aforementioned situations if we could type without looking at the keyboard and without a significant extra cognitive load.

This paper extends our previous work [1], [2] by presenting a study on a 5-key chording device that, we believe, has high potential for application in all of the above mentioned situations. This type of keyboard enables users to generate a character by simultaneously pressing a combination of keys, similarly to playing a note on a musical instrument. With five keys, there are 31 combinations in which at least one key is pressed, enough for the 26 letters of the English alphabet and five other characters. If the keys are in a position that is naturally under the fingertips, a person can type using the fingers of one hand, without committing the eyes to the keyboard. For instance, by placing the keys on the handlebar of a bike, we can control the bike with both hands and type at the same time, because typing only requires varying the pressure under the fingertips. Some visual (or auditive) feedback is still needed occasionally to verify the output and correct eventual mistakes, but this requires considerably less commitment than continuously looking at the input device. The required visual attention can be further reduced by displaying the output in the natural field of vision, for instance on a windshield or on goggles.

The main drawback of a chording device is that before being able to use it, one should learn the correspondence between key combinations and characters. We present a mapping designed to minimize the learning time by assigning intuitive combinations to each character, and a study that evaluates the proposed mapping. We will evaluate the achievable typing rates, the error rates, the characters that are more difficult to type, and the distribution of typing errors. Afterwards, we will analyze how different types of feedback (visual, auditive, and no feedback) affect the ability to type with a chording keyboard.

The paper is organized as follows. In Section II, we present a brief overview of related work. In Section III, we describe a key-to-character mapping for a 5-key keyboard designed to reduce the learning time. We denote this mapping in the following as 5keys. In Section IV, we present an experiment that evaluates the learnability of the mapping, and in Section V, we evaluate the achievable text-entry rates, typing accuracy, common error patterns, and three different feedback types. In Section VI, we conclude the paper and discuss future directions.

II. Related Work

In order to make a keyboard suitable for mobile technologies, we can make the keys very small and/or remove the one-to-one mapping between keys and characters [3]. The first method includes mini-QWERTY, on-screen keyboards, or RearType [4], and the second method includes multi-tap (with or without T9), LetterWise [5], TiltText [6], FrogPad [7], or chording keyboards. Another possibility is given by gestural text-entry techniques such as Graffiti [8], Edgewrite [9], and the minimal device independent text input method (MDITIM) [10]. Dynamic selection techniques such as Dasher [11] use probabilistic techniques to make the most likely characters or sets of characters easier to type, based on the already typed text.

Chording keyboards were first used in stenotype machines (starting from the 1830s) and telegraph communications. One such example is the Baudot code [12] (patented in 1874) that assigns five bits to a character and evolved into the International Telegraphy Alphabet No. 2, still used by some radio amateurs. Another application is represented by the Braille system, that enables blind people to read and write. Subsequent studies were performed by IBM, where researchers developed both single-handed and two-handed keyboards, with the number of keys ranging from 8 to 14 [13]. However, research stopped in 1978. Douglas Engelbart, the inventor of the computer mouse, also proposed a 5-key keyset, but this was not incorporated in any system [14]. Microwriter [15] was a chording portable word processor commercialized in the early 1980s, but again, it was not a commercial success.

As traditional desktop applications such as text editors, schedulers or e-mails have become available on mobile devices, there has been a significant increase in text entry research. This also lead to renewed interest in chording keyboards and to the appearance of several new devices: DataEgg, appeared in the early 1990s, is a 7-key handheld device with pager, phonebook, e-mail and calendar functions [16]. GKOS (2000) is a 6-key two-handed input device that can be used for text input or game control [17]. Chordite (2002), Twiddler (2004) and EkaPad (2009) are pocket sized, single-handed keyboards that can also have miniature joystick or mouse-like abilities [18], [19], [20]. The chording glove [21] is a chord keyboard where the buttons are mounted directly on the fingers. Typing studies involving chording keyboards include those performed by Lyons et al. for the Twiddler, [3], [19], [22], and by Rosberg and Slater for the chording glove [21]. Sandnes et al. propose a chording interface for controlling in-car devices such as music player, navigation system, lights or telephone [23], and an error correction mechanism for three and five-key chording keyboards [24], [25].

In mobile environments, users cannot usually look at the text-input device and/or at the display while typing; this condition is denoted as "blind" or "eyes-free" typing [26]. Therefore, it is important to analyze how visual feedback affects the text-entry process. Silfverberg examined the effect of both tactile and visual feedback when using mobile phone keypads [26] and found that reduced tactile feedback increases the typing error rate. In addition, low visual feedback also leads to more errors, decreasing accuracy. A similar study made by Clawson et al. [27], concerning typing with mini-QWERTY keyboards, demonstrates the importance of seeing the keys while typing. However, no significant differences in typing speeds and error rates were noticed when users could or could not see the typed text.

The above studies stress the importance of seeing the input device in the case of 4×3 multi-tap keypads and mini-QWERTY keyboards. But this should not be an issue for most chording keyboards, that are specifically designed to be operated without looking at the keys. Typing experiments with limited visual feedback for the Twiddler chording keyboard were performed by Lyons et al. [22], and show that, surprisingly, typing and error rates actually improve with reduced visual feedback. Mascetti et al. propose and evaluate a Braille typing system for smartphones [28]. As it is intended for visually impaired persons, there is no visual, but only audio feedback.

Other studies where participants do not look at the typing device or are involved in dynamic activities that require vision commitment include the already mentioned chording glove [21], a two-handed chorded software keyboard for PDAs [29], half-QWERTY touch typing [30], or the keyboard proposed by Gopher and Raij [31].

The chording keyboard used in this study has five keys, placed directly under the natural position of the fingertips. Unlike the Braille keyboard, it is designed to be operated by only one hand. In comparison to some of the devices presented above, with our device the users do not have to move their fingers from one key to another, so it should make no difference if they are able to see the keys or not. Considering this, we will only evaluate different feedback conditions regarding the typed text. Also, the small number of keys allows for higher design flexibility. The five keys can be directly integrated into a mobile phone case, around a computer mouse, or on a bike handlebar, thus being considered an extension rather than a separate object. This is important because some users might find it inconvenient to carry too many different devices.

III. Character Mapping

An important aspect of designing a chording keyboard is the mapping between the key combinations and the characters. One possibility is to assign easier combinations for more frequent letters, as in the Morse code, thus leading to higher typing speeds. Even if these mappings are easy to determine, the user must learn by heart the key-to-letter correspondence as there is no intuitive link between them. Another possibility is to use a semantically richer mapping, which would be easier to learn. The chording keyboard described in this paper is intended to be used in situations where desktop or other mobile keyboards are not appropriate, such as mobile environments. This is why we expect it to be used to type short texts, or to control a mobile device. Considering this, being able to easily learn the mapping is more important than being able to type fast.

We have designed the key-to-character mapping presented in this work with the primary goal of making it easy to remember. It is designed for a five-key keyboard, where each character is represented by a different key combination. From here on, we will focus only on lowercase letters, plus the period, space and backspace, as they are the most used. An additional button can be used to toggle between modes that enable typing uppercase letters, numbers, or other characters. The complete key map is given in the appendix.

To create enough possibilities for assigning an intuitive key combination to each character, we conceived five mnemonic categories. With them, a user usually can remember most combinations within minutes.

1) Single-key category: "**t**", "**i**", "**.**", "**r**", "**p**". Remembering the map for the characters in this category is totally trivial. Characters are produced by pressing a single finger and the letter is the initial of the finger. So, by pressing the key under **t**humb, **i**ndex, **r**ing and **p**inky, we obtain "**t**", "**i**", "**r**", and "**p**", respectively. There is an exception to the rule: since "**m**" fits well in another category (see below), we have reserved the middle finger for the period. The mnemonics for the characters in this category are presented in Figure 1.

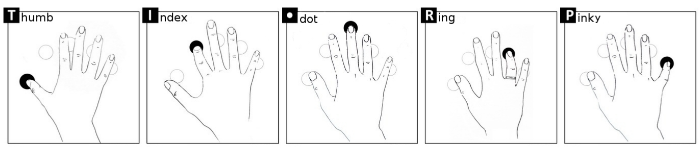

Figure 1. Single-key category

2) Fingers-down category: "**c**", "**m**", "**n**", "**u**", "**y**". The most natural way to produce the shape of a "**m**" with the hand is to stretch down the index, middle, and ring fingers, as shown in Figure 2. In a similar fashion, the shape of the fingers pressing the keys suggests the other letters in this category, namely "**c**", "**n**", "**u**", and "**y**".

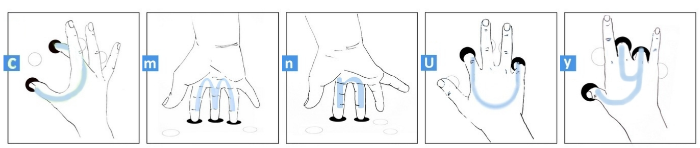

Figure 2. Fingers-down category

3) Fingers-up category: "**e**", "**l**", "**j**", "**v**", "**w**", **space**, **backspace**, **enter**. The idea is basically the same as for the previous category, but here we look at the fingers that are *not* used. A natural way to produce the shape of a "**w**" is to stretch up the index, middle, and ring fingers. The associated character is obtained by pressing the key(s) under the remaining fingers, as shown in Figure 3. "**v**", "**l**", "**e**", and "**j**", follow the same idea. We have included **space** and **backspace** in this category as **backspace** can be associated with the thumb pointing to the right and **space** with the pinky pointing to the left. For **enter**, the unused fingers represent a ∨ (pointing down) and a left arrow, suggesting the beginning of a new line.

4) Character landmark category: "**a**", "**f**", "**h**", "**k**", "**o**", "**s**", "**x**", "**z**". By looking at the shape of "**h**", we notice three landmark spots, and naturally enough, we associate them to the thumb, index and pinkie

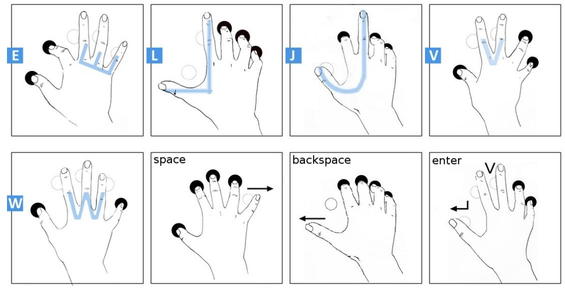

Figure 3. Fingers-up category

(see Figure 4). As a general rule, the thumb is for spots that are left and low, the index for left high, the ring for right high, and the pinky for right low. With a little bit of imagination, we can fit in this category also "**a**", "**f**", "**k**", "**o**", "**s**", "**x**", and "**z**". For "**o**", we imagine five dots spread around a circle, and we obtain it by pressing all buttons. For "**s**", we choose the points so that they remind us of a slalom (fingers-down and fingers-up alternate).

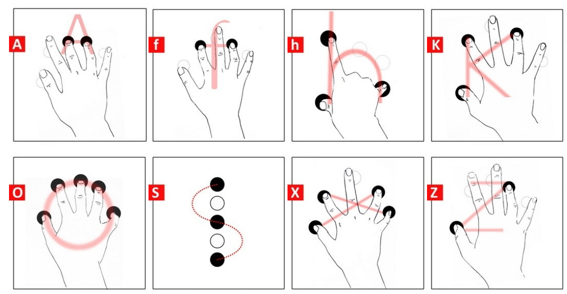

Figure 4. Character landmark category

5) Associative category: "**b**", "**d**", "**g**", "**q**". We remember these letters by associating them to similar letters. "**b**" and "**d**" can be seen as an "**o**" with a vertical bar on the left and right, respectively. We use the index and the ring fingers to represent these bars. "**g**" was inspired from "**y**" (they look alike in handwriting), and "**g**" inspires "**q**" (the tail ends left and right, respectively, so for "**g**" we use the thumb and for "**q**" the pinky). These mappings are shown in Figure 5.

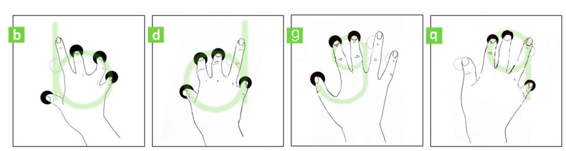

Figure 5. Associative category

The reader has probably noticed that some of the above mnemonics are easier to remember than others. With five keys, however, there are only 31 usable combinations and we use them all to map the 26 characters, plus the space, backspace, period, enter and comma. Hence, any change aimed at improving one mnemonic implies at least one other change.

The effectiveness of the proposed mapping is assessed through the studies described in the next sections. In the first study, we compare this mapping to two others from a learnability point of view, or how easy users can remember the key-to-character correspondences. In the second study, we estimate the usability of the mapping (typing speed, accuracy [32] and the most common mistakes), and how three different types of feedback affect the typing process.

IV. Learnability Study

This first study compares, from a learnability point of view, the proposed mapping (5keys) to two others. The references are the Microwriter mapping [15], also based on intuitive mnemonics, and the Baudot code [12], which is based on letter frequency and assigns easier key combinations to the most common characters. All three mappings are designed for five-key keyboards.

A. Experimental Setup

Design: The experiment had a 3×10 between-subjects design. Each of the 30 participants was assigned to work under one of the three conditions: 5keys, Microwriter or Baudot.

The experiment consisted of three sessions of three rounds each. For each participant, the sessions took place on consecutive days. For each round, the subjects had 5 minutes to look at a printed version of the mappings and try to remember them. Afterwards, they used a Java application to warm up, by typing each letter of the alphabet. During the warm-up phase, a help image showing the key combination for the letter to be typed was shown to the participants. In the next step, the help image was not available any more and the participants had to type the alphabet three times. The order of the letters was random, but the same for all participants. The subjects had five seconds and only one attempt to type each target character. The correct key combination was displayed for one second when the user typed a character (right or wrong), or when the user typed nothing for five seconds. The typing rounds were separated by breaks of two minutes.

Participants: We have recruited 30 participants, 10 for each of the three mappings, from the students of our university (undergraduate, master's and PhD programs). The participants were between 19 and 30 years old, and four were female. None of the subjects had used a chording keyboard before. As the participants who know how to play a musical instrument could have had an advantage, they were equally distributed among the three experiment groups. We also tried to equally distribute them based on gender and study level. Two participants abandoned the experiment after the first session, one testing the 5keys and one the Microwriter mapping.

Equipment and Software: We designed a Java application to simulate the chording keyboard on a regular QWERTY Apple desktop keyboard. It only allows for the use of five keys, each representing a key of the chording keyboard. Each of these keys corresponds to a finger of the right hand. A screenshot of the application is visible in Figure 6. The top-left window contains the target characters to be typed. The bottom-left window represents the typing area, and the help image is displayed on the right.

The Java application recorded log files containing the time of each key press and release, the typed text, the corresponding key combination, the total number of errors and the total time spent writing each character.

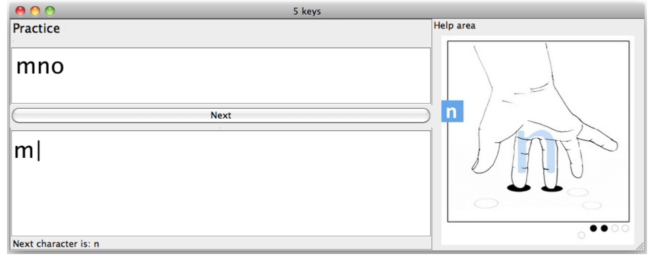

Figure 6. Application interface used during the study

Procedure: The participants were given written and verbal instructions regarding the goal of the experiment. The participants were given unique anonymous ID and were shown only the mapping that they were going to use. For each session of the experiment (approximately 30 minutes), they received a fixed monetary compensation.

During the experiment, the participants sat at a desk. Before the first session, the participants were explained how to press multiple keys to generate the chords, and were allowed to choose which five keys of the desktop keyboard they wanted to use. The only constraint was the space key for the thumb. They were not able to change the keys afterwards. A typical choice was the keys for f, t, y and u for the index, middle finger, ring and pinky, respectively.

The software was self-administered. Once started, it launches the warm-up phase, and then it goes automatically to the typing phase. The characters to be typed are also updated automatically.

B. Experiment Results

For each typing round, the participants had to type a total of 78 characters (3×26). To determine which of the mappings is easier to learn, we compared the number of errors (wrongly typed or not typed characters) for each round. Exponential regressions were derived to fit these error values. The average values for each mapping and for each round, and the exponential regressions are presented in Figure 7.

After two sessions (six rounds of approximately five minutes of typing each), the total number of errors was considerably lower for the mnemonic based mappings (5keys and Microwriter) compared to the mapping based on letter frequency. Therefore, we conclude that mnemonic based mappings are learned faster. This is confirmed by the anova test ($F = 24.15$, $p = 0.0001$). The goal of the study was to evaluate which mapping is easier to learn, and the Baudot mapping is clearly more difficult. Hence, in the third session, we only analyzed the 5keys and Microwriter mappings. Upon checking the average number of errors, no significant difference between these two was noticed ($F = 0.95$, $p = 0.358$). An advantage of 5keys can be observed from the analysis of the regression curves, because the curve for 5keys is slightly below the curve for Microwriter.

In addition to the total number of errors, the number of characters that were typed wrong at least once for each round

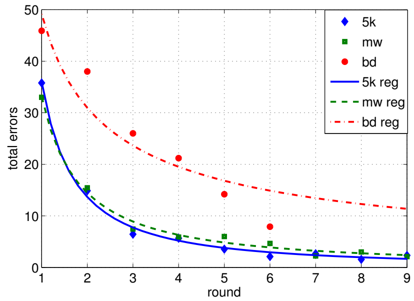

Figure 7. Average number of errors (for each mapping and for each round) and regression curves

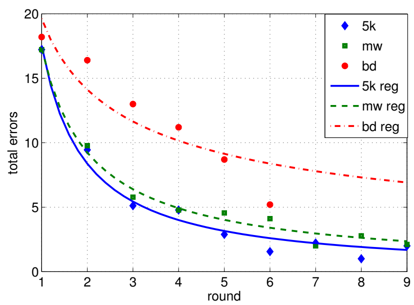

Figure 8. Average number of character errors (for each mapping and for each round) and regression curves

were also compared. For example, if "**a**" was typed incorrectly two times, this counted only as one character error. The results are shown in Figure 8 and provide an indication of how many characters are difficult to remember for each mapping. In this case, the difference between the 5keys and Microwriter mappings is more visible than in Figure 7, and statistically significant ($F = 5.4$, $p = 0.0486$). As expected, both mnemonic-based mappings lead to significantly less character errors than the Baudot mapping ($F = 18.65$, $p = 0.0004$). The Baudot mapping might lead to higher typing rates, but ascertaining this was not the goal of the presented study.

At the end of the third typing session (after nine rounds or approximately 45 minutes of actual typing), the participants were asked how confident they felt about their knowledge of the mappings and if they could use the presented method as a text input mechanism. All of them answered affirmatively and most mentioned that they had completely learned the mappings. This is confirmed by a low error rate (3.16% after 6 rounds and 2.14% after 9 rounds for the proposed mapping).

From this experiment, we draw the conclusion that a mnemonic-based mapping facilitates the process of learning the code. We also conclude that the proposed 5keys mapping outperforms the Microwriter mapping, also mnemonic-based, in terms of average error rate.

The mnemonic set was designed based on the finger positions of the right hand. Two of the participants (one for the 5keys and one for the Microwriter) were left-handed. Yet, they also typed with their right hand and, interestingly, their error rates were actually lower than the average.

V. Usability Study

The first study showed that an intuitive mapping can be learned in less than 45 minutes. It was followed by an independent experiment aimed at determining achievable typing rates, accuracy, and common error patterns for the 5keys mapping. Moreover, we evaluated different feedback types, when the text can and cannot be seen.

The input method that we present is designed to be used in situations where the visual attention is partially or totally unavailable for the typing process. In these conditions, audio feedback is often suggested as an alternative. This is indeed useful in some environments, but could be difficult to use in noisy areas. Considering this, we designed a 3×10 within-subjects experiment where we analyzed three different typing conditions. Under the first condition, subjects were able to see the outcome of what they have typed, under the second condition they received voice output for each typed letter (without visual feedback), and under the third condition they received no feedback at all about the typing. From here on, we will refer to these conditions as visual, auditive, and no-feedback, respectively.

A. Experimental Setup

Although there are a few similar aspects between this experiment and the previous one (same mapping and similar software interface), the study structure is completely different. Therefore, in this subsection, we will describe the new experimental setup.

Design: The experiment had a 3×10 within-subjects design. Each of the ten participants was asked to type under all three conditions.

The experiment was based on a Java application similar to the one shown in Figure 6, but the subjects were asked to type full sentences and used a chording keyboard prototype, not a desktop keyboard. The participants were asked to type for 10 sessions of 30 minutes. Each session consisted of three rounds of 10 minutes separated by breaks of 2 minutes, and each round corresponded to a different typing condition. The order of the typing conditions was random for each session, but the same for all subjects. For each user, the typing sessions took place on consecutive days, with the exception of weekends.

The first session enabled the subjects to remember the mapping between keys and characters. A help image showing the key combination for the letter to be typed was always displayed. During the subsequent sessions, the help image was only available on demand by pressing the *shift* key. At the beginning of each round, the participants warmed up by typing each letter of the alphabet. Afterwards, they typed phrases from a set considered representative of the English language [33]. These phrases were pre-prepared before the experiment to contain only small letters and no punctuation signs.

Participants: Ten participants took part in this study. Six of them also took part in the learnability study described in Section IV, using the 5keys mapping. The other four participants did the same experiment on a different occasion. Overall, they had approximately 45 minutes of training. All were PhD students from our university, eight male, two female, between 24 and 31 years old.

Equipment and Software: The keyboard prototype has the keys placed around a computer mouse and is presented in Figure 9. We designed the prototype in this way because we wanted the subjects to see a practical application of a chording device: allowing typing and screen navigation at the same time, with only one hand. The buttons are placed so that they can be easily operated while holding the mouse with the palm. We used keys and not pressure or touch sensors because they provide a distinct tactile feedback. The keyboard is designed using an Arduino Pro Mini microcontroller board and communicates with the computer by Bluetooth. The buttons are placed in a position that is naturally under the fingertips when the users hold their palm on the mouse.

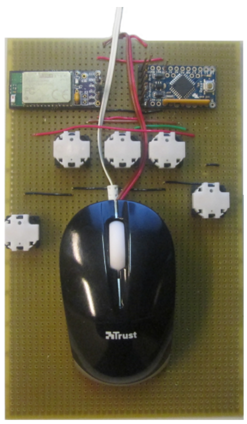

Figure 9. Chording keyboard prototype

The Java application is similar to the one described in Section IV. The typed text was displayed only for the visual condition. For the auditive condition, the participants used headphones to receive feedback. Log files containing the time of each key press and release, the typed text, the number of occurrences for each character, the corresponding key combination, the total number of errors and the total time spent writing each character were recorded. For each typing error, we checked what character was typed in lieu of the correct one.

Procedure: As before, the participants were given written and verbal instructions regarding the goal of the experiment. During the experiment, they sat at a desk, and each of them had an unique anonymous ID. The participants were instructed to type as quickly and as accurately as possible. Once a target text was completed, they were instructed to press the 'Next' button in order to display the next target text. They were told to not correct eventual mistakes and to keep typing, but this was not enforced and they were allowed to delete typed text. As a reward for the time commitment during the experiment, they received a fixed monetary compensation for the first nine sessions. For additional motivation, for the last session, the reward was proportional to the number of typed words and to the accuracy.

Experimental Data: The total amount of data gathered during the experiment consists of $40\,345$ words, out of which 4052 (10.17%) contain errors. The total number of characters is $219\,308$, from which 6386 (2.91%) are errors.

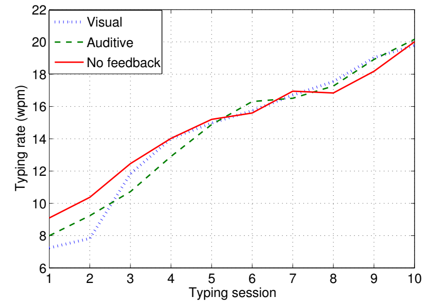

Figure 10. Average typing rates for each condition and for each typing session

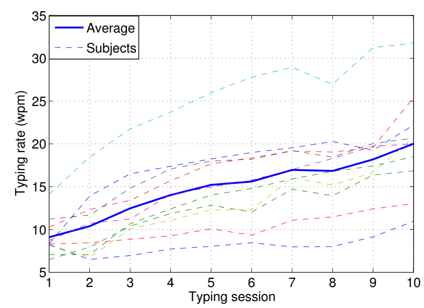

Figure 11. Average typing rates for each subject and for each typing session, for the no-feedback condition

B. Text-Entry Speed

We use the words-per-minute measure to describe the text-entry speed. This is defined as

$$wpm = \frac{60L}{t}\frac{1}{5}, \tag{1}$$

where L is the total number of typed characters and t is the typing time in seconds. The scaling factor of 1/5 is based on the fact that the average English word length is approximately 5 characters. Because the average word length for the typed text differed from one session to another, the use of the above formula provides a more reliable estimate than actually counting the words.

In Figure 10, we show the average typing rates for each session and for each condition. For the first three sessions, the rates are higher for the no-feedback condition, and the anova tests showed that the differences are statistically significant ($F = 10.85$, $p < 0.0001$). From the fourth session onward, the differences between the typing rates are not so visible. Moreover, the effect of the feedback type is no longer significant ($F = 0.28$, $p = 0.75$). This probably happened because, in the beginning, subjects paused while typing to check the provided feedback, visual or audio. As they gained experience, they became more confident and did not analyze the feedback as often, therefore reducing the differences between conditions.

In Figure 11, we show the typing rates for each user and for each session, during the no-feedback condition. We notice that the fastest subject typed three times faster than the slowest subject, the differences being statistically significant ($F = 53.8$, $p < 0.0001$).

Figure 12. Chording keyboard prototype mounted on a bike handlebar. The four visible buttons correspond to the index, middle, ring and pinky fingers. The thumb button is on the other side of the prototype.

At the end of the experiment, the average typing rates were 19.77, 20.16 and 20.00 wpm for the visual, audio and no-feedback conditions, with maximums of 31.24, 30.48 and 31.78 wpm, respectively. Considering the participants' experience from the previous experiment, these values correspond to approximately 350 minutes of practice. Because the text entry rates would probably still improve, we use exponential regressions to estimate how fast people will be able to type after longer training periods. Based on these calculations, after 20 sessions (300 more minutes of practice), the average could be 26 wpm, and the fastest typist could reach 42 wpm.

As a reference, the typing rates achieved after 350 minutes of practice are 13.5 wpm for multi-tap mobile phones [5] and 24.2 wpm for Twiddler [19]. Rates of 20.36 wpm were reached by expert T9 users [34]. Handwriting speed is usually between 15 and 25 wpm [35]. We point out that the experimental conditions were not the same for all devices, hence, the above typing rates are only of indicative nature. For both multi-tap and T9 techniques, visual attention is essential for most users. For the 5keys device, it makes essentially no difference if the user has visual contact with the keys or not. It should also be taken into consideration that Twiddler uses 12 keys, whereas our mapping only requires 5 keys, thus providing a clear space advantage and more design flexibility. If placed in a position that is naturally under the fingertips (for example on the handlebar of a bike, as in the prototype from Figure 12), the users will have continuous access to the keys. Moreover, users do not have to move their fingers from one key to another, which probably leads to fewer errors.

C. Error Analysis

We used the total error metric presented by Soukoreff and Mackenzie in [36]: it considers both corrected and uncorrected errors. It is defined as

$$ErrorRate = \frac{IF + INF}{C + IF + INF} \times 100\% \,, \qquad (2)$$

where C is the number of correctly typed characters, INF is the number of incorrectly spelled characters that were not corrected (it includes substitutions, when one character is typed for another, insertions, when an extra character is typed, and deletions, when a character is omitted), and IF is the number of incorrectly spelled characters that were corrected by the user.

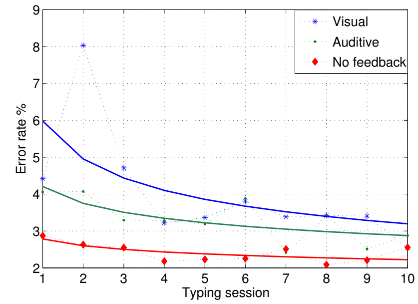

Figure 13. Average error rates and regressions for each condition and for each typing session

The errors could have two main causes: the subject does not recall the correct key combination or, alternatively, a coordination mistake is produced during execution. We call these error types cognitive and sensorimotor errors, respectively. We expect the cognitive errors to decrease faster, as a function of training, because it is easier to learn the code than to improve motor skills. This is confirmed by the statements of the participants in the two experiments: they said that they had learned the mapping by the end of the training, and the errors were due to lack of attention or finger combinations that seemed more difficult.

In Figure 13, we display the average error rates for each session, accounting for both uncorrected and corrected errors, and the corresponding exponential regressions. All of the error rates are below 5%, except for the second typing session, visual condition. The reason for this could be the fact that in the first session the help image was always displayed, whereas in the second session it was hidden. Moreover, the first typing condition in session 2 was the visual one, giving subjects more practice time for the auditive and no-feedback conditions, which do not have much of an increase in the error rates. The averages for all sessions and for all users under the visual/auditive/no-feedback conditions are 3.41%, 2.97% and 2.32%, respectively. Anova tests show that feedback plays a relevant role in the error rates ($F = 25.57$, $p < 0.0001$).

Initially, it might seem surprising that the error rates are the lowest for the no-feedback condition and the highest for the visual condition. This is explained, however, by the fact that increased cognitive loads generally lead to more errors [37]. For our study, the cognitive load is the highest in the visual condition: users can check the whole typed phrase; it is reduced by the audio condition when users only hear the last typed character, and minimum in the absence of feedback. Noticing an error could cause someone to become less focused, thus favoring new mistakes.

The error rates decrease during the first four sessions (with the exception mentioned above), but afterwards they remain stationary or even increase. Similar effects, when after a certain point the error rates do not decrease anymore as users gain experience, are also noticed by Matias et al. [30] and Lyons et al. [19].

As in Section V-B, we compare the error rates with those

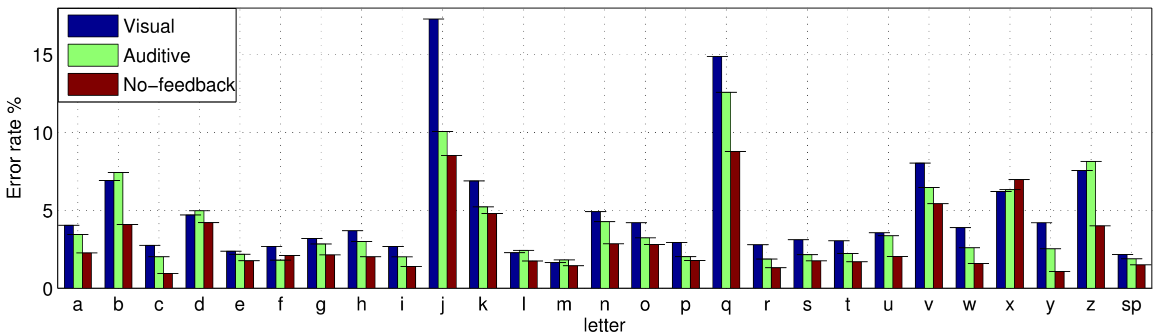

Figure 14. Error rates for each character for each typing condition

TABLE I. AVERAGE ERROR RATES(%) VERSUS THE REQUIRED NUMBER OF KEYS

Typing condition	Number of keys					Anova test
	1	2	3	4	5	
All conditions	2.06	3.01	4.99	4.73	3.40	$F = 1.15$, $p = 0.358$
Visual	2.74	3.89	6.39	5.00	4.20	$F = 0.92$, $p = 0.472$
Auditive	1.91	3.18	4.87	5.14	3.22	$F = 1.24$, $p = 0.323$
No-feedback	1.50	2.14	3.67	4.19	2.81	$F = 1.46$, $p = 0.248$

TABLE II. AVERAGE ERROR RATES(%) VERSUS THE MNEMONIC CATEGORY. SK - SINGLE-KEY; FD - FINGERS-DOWN; FU - FINGERS-UP; CL - CHARACTER LANDMARK; A - ASSOCIATIVE.

Typing condition	Mnemonic category					Anova test
	SK	FU	FD	CL	A	
All conditions	2.06	2.53	4.51	3.63	6.31	$F = 1.15$, $p = 0.358$
Visual	2.74	3.29	5.83	4.77	7.43	$F = 0.92$, $p = 0.472$
Auditive	1.91	2.72	4.21	3.91	6.94	$F = 1.24$, $p = 0.323$
No-feedback	1.50	1.63	3.28	3.30	4.80	$F = 1.46$, $p = 0.248$

for multi-tap (∼5%) and Twiddler (4.2%) after 350 minutes of practice, and also with expert T9 users (0.52%). Even if the Twiddler allows for higher typing rates, the error rates are also higher. Again, these values are only indicative, due to different experimental conditions.

It is important to analyze the error rates for each character, because this knowledge could be used to design a more efficient learning technique for the proposed mapping. For instance, subjects could be asked to practice more on characters with higher error rates. In Figure 14, we present the error rates for each character and for each typing condition. We notice that the character errors respect the pattern of the overall error rates: the highest for the visual condition and the lowest for the no-feedback condition: this is the case for 20 of the 27 analyzed characters.

For all three conditions, the error rates are higher for characters that are less frequent in the English language, such as "**q**" and "**j**", probably because the subjects had fewer opportunities to practice on them. Non-negligible error rates can also be observed for high-frequency characters, as users probably try to type faster as they gain more experience. The character error rates are similar between the three conditions, up to a scaling factor: if a character has an error rate lower than other characters for a specific condition, it usually also has a lower error rate relative to the same other characters for the other conditions. This is confirmed by the correlation coefficients between the error vectors, all above 0.9.

TABLE III. MOST FREQUENT LETTER SUBSTITUTIONS

substitution	percentage	substitution	percentage
$v \to b$	3.21	$n \to a$	1.88
$q \to j$	2.90	$q \to p$	1.69
$q \to d$	2.90	$j \to f$	1.61
$b \to y$	2.46	$x \to o$	1.49
$j \to x$	2.42	$d \to g$	1.42
$j \to q$	2.42	$v \to z$	1.35
$k \to h$	2.24	$p \to r$	1.34

In general, characters requiring three or four keys have higher error rates, but this is not statistically significant, as shown in Table I. Letters from the single-key category have the lowest error rates and those from the associative category the highest, but again, these results are not statistically significant (Table II).

D. Common Errors

To understand the error patterns that appear most frequently, we computed the confusion matrix [38] corresponding to the typed text. This is a square matrix with rows and columns labeled with all possible characters. The value at position *ij* shows the frequency of character *j* being typed when *i* was intended. The values are given as percentages from the total number of occurrences for character *i*.

In Table III, we present the 14 most common substitutions and the corresponding percentages. The values correspond to the whole experiment, including all three typing conditions. The confusion matrices for the whole experiment and for each condition are similar, with correlation coefficients higher than 0.99. If we consider only the erroneously typed characters (by setting the diagonal values, which are at least two orders of magnitude higher than the other values, to zero), the correlation coefficients are above 0.9, still showing a strong similarity: if one character is frequently typed instead of another under one condition, the same will happen under the other two conditions; if the probability for one character to be typed instead of another is low under one condition, it is also low under the other two conditions.

It is useful to represent a key combination by a 5-bit codeword in which the first digit represents the key under the thumb, the second digit the key under the index, etc. The value of a position is 1 if the corresponding key is pressed. So, for instance, 10111 is the codeword for "**b**", for which all fingers, except the index, press the keys. By analyzing the 5-

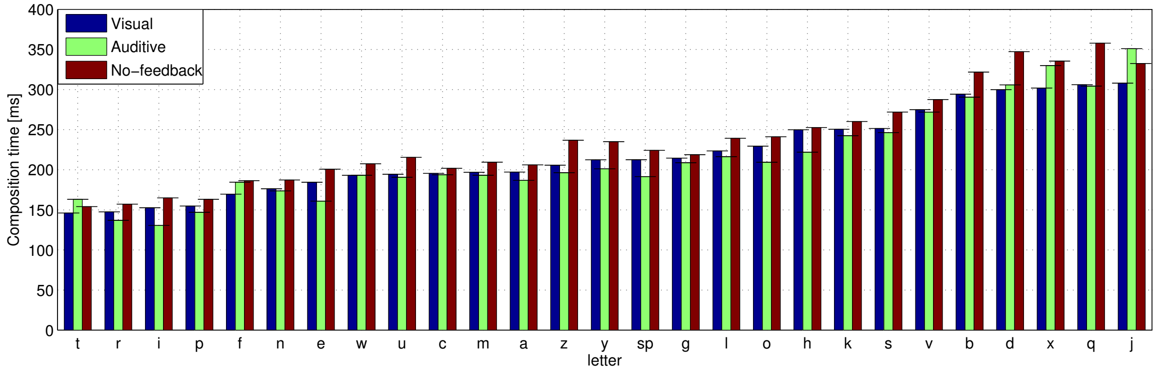

Figure 15. Composition times for each character for each typing condition

bit code for the substitutions from Table III, we notice that in 8 of the 14 cases the errors appear between characters that differ only by one bit (for example "**b**", code 10111, and "**v**", code 10011). In the other six cases, the errors appear between characters that differ by two bits. Overall, 44.81% of the wrongly typed characters differ from the intended character by one bit, 39.22% by two bits, 8.72% by three bits, 5.87% by four bits and 1.39% by five bits.

If we check word by word and consider only substitution errors, i.e., errors that arise from substituting individual characters (75% of the total errors), 91.48% of the erroneous words contain one substitution, 7.68% contain two substitutions and 0.67% three substitutions. From a bit-error point of view, 40.56% of the erroneous words contain a one-bit error, 40.92% a two-bit error, 7.62% a three-bit error, 7.20% a four-bit error and 2.13% a five-bit error. These values were used to implement an error correcting mechanism that relies both on a dictionary and on the probability that a character be substituted for another [39].

E. Character Typing Duration

As the coordination effort is not the same for all key combinations, we expect that different characters require more time than others to be typed. The time needed to form a key combination, called composition time, is measured from the moment the first key of a combination is pressed until a key is released. It is when a key of the combination is released that the corresponding character is produced. From that moment on, the pressing of a key indicates the start of a new character. In Figure 15, we present the average composition time for each character. Instead of ordering the characters alphabetically, we order them with respect to the composition time under the visual condition.

We notice that the composition times are higher for characters which are less frequent in the English language, such as "**q**" and "**j**", or for characters that require four keys, as "**x**", "**d**" and "**b**". Not surprisingly, "**o**" requires less time than the four-key letters, as it is easier to press all five keys than to press a specific subset of them. Also, letters requiring key combinations perceived as more difficult (for example "**q**", code 01101, or "**d**", code 11101, for which the middle finger and the pinky are down while the ring finger is up) require more time than others. In general, the

TABLE IV. AVERAGE COMPOSITION TIMES EXPRESSED IN MILLISECONDS VERSUS THE REQUIRED NUMBER OF KEYS

Typing condition	Number of keys					Anova test
	1	2	3	4	5	
All conditions	151	194	254	288	277	$F = 10.54$, $p < 0.0001$
Visual	150	189	248	277	229	$F = 13.31$, $p < 0.0001$
Auditive	144	184	245	279	209	$F = 8.25$, $p = 0.0003$
No-feedback	159	205	266	307	241	$F = 10.35$, $p < 0.0001$

TABLE V. AVERAGE COMPOSITION TIMES EXPRESSED IN MILLISECONDS VERSUS THE MNEMONIC CATEGORY. SK - SINGLE-KEY; FD - FINGERS-DOWN; FU - FINGERS-UP; CL - CHARACTER LANDMARK; A - ASSOCIATIVE

Typing condition	Mnemonic category					Anova test
	SK	FU	FD	CL	A	
All conditions	151	198	237	236	290	$F = 6.45$, $p = 0.0014$
Visual	150	195	232	231	278	$F = 7.01$, $p = 0.0008$
Auditive	144	190	230	227	277	$F = 4.77$, $p = 0.0064$
No-feedback	159	209	248	248	311	$F = 6.96$, $p = 0.0009$

composition time increases with the number of required keys per character, and the dependence is statistically significant. This is shown in Table IV. We also studied the dependency between composition times and the letter category and we summarize the results in Table V. These values confirm those from Table IV, as the average number of keys per character is 1 for the single-key category, 2.4 for fingers-down, 2.8 for fingers-up, 3 for character landmark, and 3.5 for the associative category. These results are also statistically significant.

As subjects gained more experience, they were able to type faster and the average letter duration decreased from 265.9 milliseconds in the first session under the visual condition to 183.9 milliseconds in session 10, or by 29.5%. During the same period, the text entry rates increased from 7.23 to 19.78 wpm, or by 173.8%. The difference is explained by the fact that the idle time between the end of one character and the beginning of the next also decreased.

VI. CONCLUSION

In this paper, we have presented the results of a study aimed at evaluating a mapping for a chording input device. The overhead needed to learn the mapping was reduced by choosing easy-to-remember key combinations. A first experiment showed that the mapping was learned after less than 45 minutes of actual typing. Moreover, the total number of errors was considerably smaller than for a letter-frequency based mapping and slightly smaller than for another mnemonic based mapping.

A second experiment enabled us to determine achievable typing rates, error rates, and the effect of different types of feedback for a chording keyboard. The subjects were asked to type under three conditions: with visual feedback, with auditive feedback and with no feedback at all. Due to the keyboard design, whether the user can see the keys or not should not make any difference on the typing process — at the end of the experiment, participants confirmed that they did not look at the keys. Similarly, someone playing a saxophone does not look at the keys to be pressed.

After approximately 350 minutes of typing (taking into consideration the previous typing experience of the subjects), the average entry rates are approximately 20 wpm under all three conditions, with the maximums above 30 wpm. We conclude, therefore, that having visual, audio or no feedback has no influence on the typing speed with the presented chording keyboard. The average error rates are 2.32% under the no-feedback, 2.97% under the auditive and 3.41% under the visual conditions. This is explained by the fact that the cognitive loads are different under the three typing conditions: the highest under the visual and the lowest under the no-feedback condition. Hence, not seeing the typed text actually provides an advantage. The error patterns are similar between conditions, the characters with the highest error rates and the most common substitutions being the same. We also analyzed which characters are perceived as more difficult to type and the most common errors. This data was used to develop an error correction mechanism specifically designed for a chording keyboard using the proposed mapping.

During the experiments, the subjects sat at a desk. To go one step further, we designed, built and tested a prototype for a bike, shown in Figure 12. We fit the five keys under the natural position of the fingers on the handlebar. With the help of a wrapper application that captured the input text, we used the keyboard to control the operation of a smartphone: controlling the music player, writing a short note, or initiating a phone call, without touching or looking at the touchscreen. Two of the authors tested the device and felt that they could effortlessly ride and type while controlling the bike with both hands and staying focused on the road. Moreover, as the keys are directly under the fingers, we could also type accurately on a slightly bumpy road. Though encouraging, these results are only exploratory, and performing a more detailed study will be difficult due to legal issues related to the risk of accidents.

The presented text entry method is not designed to completely replace desktop or on-screen keyboards, but to be used in certain specific situations. The typing rates comparable to handwriting speed, the low error rate, and the fact that the lack of visual or audio feedback does not impede the typing process make it a valuable option for situations where a person is not able to continuously check the output. In addition, the keyboard can be used with only one hand. The small number of keys also represents an advantage from the size and design flexibility point of view. Even if so far we envisaged the chording keyboard as a means of typing in dynamic or busy environments, due to its advantages, it can also be successfully used in other areas: for example, it can facilitate text input for disabled users who can only use one hand, or for persons who are visually impaired.

APPENDIX

During the study, we have only focused on small letters and space. However, with the help of a mode button, we can also type capitals, numbers and other symbols. The complete mapping is given in Table VI. The numbers are given by translating the code from binary to decimal. The least significant bit is the left one, because it is probably easier to think of the thumb as "one" and index as "two" than of the pinky as "one" and ring as "two". For other symbols, we tried to provide logical correspondences, such as the same code for "**a**" and "@", for "**m**" from minus and "**-**", for "**u**" from underscore and "_", etc. Other mappings may be less intuitive, but it is virtually impossible to assign easy-to-remember correspondences to all characters. Several symbols repeat, and this could be used to accommodate other characters, such as letters with diacritics.

TABLE VI. COMPLETE MAPPING

Five bit code	Character			
	Mode 1	Mode 2	Mode 3	Mode 4
00110	a	A	∼	@
10111	b	B	]	}
10100	c	C	5	(
11101	d	D	)	)
11000	e	E	3	=
01010	f	F	”	:
11100	g	G	7	&
11001	h	H	>	>
01000	i	I	2	up
01011	j	J	:	$
11010	k	K	*	ˆ
00111	l	L	\	/
01110	m	M	;	-
01100	n	N	6	—
11111	o	O	0	*
00001	p	P	(	right
01101	q	Q	=	”
00010	r	R	8	down
10101	s	S	+	+
10000	t	T	1	left
01001	u	U	’	_
10011	v	V	tab	tab
10001	w	W	¡	¡
11011	x	X	[	{
10110	y	Y	@	#
10010	z	Z	9	%
11110	space			
01111	backspace			
00011	new line			
00100	.	?	4	’
00101	,	!	-	;

REFERENCES

[1] A. Tarniceriu, P. Dillenbourg, and B. Rimoldi, "The effect of feedback on chord typing," in The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '13, Porto, Portugal, September 2013, pp. 69–74.

[2] A. Tarniceriu, P. Dillenbourg, and B. Rimoldi, "Single-handed typing with minimal eye commitment: A text-entry study," in The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '12, Barcelona, Spain, September 2012, pp. 117–122.

[3] K. Lyons and R. Catrambone, "Improving novice performance on the twiddler one-handed chording keyboard," in Proceedings of the International Forum on Applied Wearable Computing, IFAWC '05, 2005, pp. 145–160.

[4] J. Scott, S. Izadi, L. S. Rezai, D. Ruszkowski, X. Bi, and R. Balakrishnan, "Reartype: text entry using keys on the back of a device," in Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI '10. ACM, 2010, pp. 171–180.

[5] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner, "Letterwise: prefix-based disambiguation for mobile text input," in Proceedings of the 14th annual ACM symposium on User interface software and technology, ser. UIST '01. Orlando, Florida, United States. ACM, 2001, pp. 111–120.

[6] D. Wigdor and R. Balakrishnan, "Tilttext: using tilt for text input to mobile phones," in Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST '03. ACM, 2003, pp. 81–90.

[7] URL: http://www.frogpad.com/ [accessed: 2014-05-12].

[8] D. Goldberg and C. Richardson, "Touch-typing with a stylus," in Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems, CHI '93. ACM, 1993, pp. 80–87.

[9] J. O. Wobbrock, B. A. Myers, and J. A. Kembel, "Edgewrite: a stylus-based text entry method designed for high accuracy and stability of motion," in Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST '03. ACM, 2003, pp. 61–70.

[10] P. Isokoski and R. Raisamo, "Device independent text input: a rationale and an example," in Proceedings of the working conference on Advanced visual interfaces, AVI '00. New York, NY, USA: ACM, 2000, pp. 76–83.

[11] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, "Dasher a data entry interface using continuous gestures and language models," in Proceedings of the 13th annual ACM symposium on User interface software and technology, ser. UIST '00. New York, NY, USA: ACM, 2000, pp. 129–137.

[12] A. Ralston and E. D. Reilly, Eds., Encyclopedia of computer science (3rd ed.). Van Nostrand Reinhold Co., 1993.

[13] F. C. Bequaert and N. Rochester, "Teaching typing on a chord keyboard," IBM Technical Report, Tech. Rep., 1977.

[14] D. C. Engelbart, "Design considerations for knowledge workshop terminals," in Proceedings of the June 4-8, 1973, national computer conference and exposition, AFIPS '73. ACM, 1973, pp. 221–227.

[15] URL: http://www.ericlindsay.com/palmtop/mwrite.htm/ [accessed: 2014-05-12].

[16] URL: http://www.xaphoon.com/dataegg/ [accessed: 2014-05-12].

[17] URL: http://gkos.com/ [accessed: 2014-05-12].

[18] URL: http://chordite.com/ [accessed: 2014-05-12].

[19] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney, "Twiddler typing: one-handed chording text entry for mobile phones," in Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '04. Vienna, Austria: ACM, 2004, pp. 671–678.

[20] URL: http://www.ekatetra.com/products/ekapad.html [accessed: 2014-05-12].

[21] R. Rosenberg and M. Slater, "The chording glove: a glove-based text input device," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 29, no. 2, 1999, pp. 186 –191.

[22] K. Lyons, D. Plaisted, and T. Starner, "Expert chording text entry on the twiddler one-handed keyboard," in Proceedings of the Eighth International Symposium on Wearable Computers, ISWC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 94–101.

[23] F. E. Sandnes, Y.-P. Huang, and Y.-M. Huang, "Near eyes-free chauffeur computer interaction with chording and visual text mnemonics," J. UCS, vol. 16, no. 10, 2010, pp. 1311–1326.

[24] F. E. Sandnes and Y.-P. Huang, "Chording with spatial mnemonics: Automatic error correction for eyes-free text entry," J. Inf. Sci. Eng., vol. 22, no. 5, 2006, pp. 1015–1031.

[25] F. Sandnes and Y.-P. Huang, "Non-intrusive error-correction of text input chords: a language model approach," in Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American, june 2005, pp. 373 – 378.

[26] M. Silfverberg, "Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices," in Human-Computer Interaction with Mobile Devices and Services, Lecture Notes in Computer Science, L. Chittaro, Ed. Springer Berlin Heidelberg, 2003, vol. 2795, pp. 76–90.

[27] J. Clawson, K. Lyons, T. Starner, and E. Clarkson, "The impacts of limited visual feedback on mobile text entry for the twiddler and mini-qwerty keyboards," in Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on, oct. 2005, pp. 170 – 177.

[28] S. Mascetti, C. Bernareggi, and M. Belotti, "Typeinbraille: Quick eyes-free typing on smartphones," in Computers Helping People with Special Needs, ser. Lecture Notes in Computer Science, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds. Springer Berlin Heidelberg, 2012, vol. 7383, pp. 615–622.

[29] K. Yatani and K. N. Truong, "An evaluation of stylus-based text entry methods on handheld devices studied in different user mobility states," Pervasive and Mobile Computing, vol. 5, no. 5, 2009, pp. 496 – 508.

[30] E. Matias, I. S. MacKenzie, and W. Buxton, "One-handed touch typing on a qwerty keyboard," Hum.-Comput. Interact., vol. 11, no. 1, Mar. 1996, pp. 1–27.

[31] D. Gopher and D. Raij, "Typing with a two-hand chord keyboard: will the qwerty become obsolete?" Systems, Man and Cybernetics, IEEE Transactions on, vol. 18, no. 4, july-aug. 1988, pp. 601 –609.

[32] R. W. Soukoreff, "Text entry for mobile systems: Models, measures, and analyses for text entry research," Master's thesis, York University, 2002.

[33] I. S. Mackenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in Extended Abstracts of the ACM Conference on Human Factors in Computing Systems CHI '03. Fort Lauderdale, Florida, United States: ACM, 2003, pp. 766–767.

[34] C. L. James and K. M. Reischel, "Text input for mobile devices: comparing model prediction to actual performance," in Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01. Seattle, Washington, United States: ACM, 2001, pp. 365–371.

[35] I. S. MacKenzie and R. W. Soukoreff, "Text Entry for Mobile Computing: Models and Methods, Theory and Practice," Human-computer Interaction, vol. 17, 2002, pp. 147–198.

[36] R. W. Soukoreff and I. S. MacKenzie, "Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03. New York, NY, USA: ACM, 2003, pp. 113–120.

[37] A. Baddeley, Working Memory, Oxford Psychology Series, No 11. Clarendon Press, 1986.

[38] K. Kukich, "Techniques for automatically correcting words in text," ACM Comput. Surv., vol. 24, December 1992, pp. 377–439.

[39] A. Tarniceriu, B. Rimoldi, and P. Dillenbourg, "Error correction mechanism for five-key chording keyboards," in The 7th International Conference on Speech Technology and Human-Computer Dialogue, SpeD '13, Cluj-Napoca, Romania, October 2013.