

Gestyboard 2.0: A gesture-based text entry concept for high performance ten-finger touch-typing and blind typing on touchscreens

Tayfur Coskun¹, Christian Wiesner¹, Eva Artinger¹, Amal Benzina¹, Patrick Maier¹,
Manuel Huber¹, Claudia Grill¹, Philip Schmitt¹, and Gudrun Klinker

Anonymous
anonymous
WWW home page anonymous

Abstract. This paper presents the second version of the **Gestyboard**, which is an innovative approach of text entry on multi-touch devices like tabletops or tablets. To overcome the lack of tactile feedback, we use unique gesture-to-key mappings for each finger according to the ten-finger touch-typing method. As a key feature, the Gestyboard only accepts keystrokes when they are performed with the finger corresponding to the ten-finger touch-typing method. This way, missing a keystroke is not possible, and therefore blind typing is naturally supported by the concept. The first version of the Gestyboard was optimized according to the qualitative and quantitative results of our first formal evaluation. This paper presents two new evaluations which give new insights on the comparative performance and conceptual improvements of the Gestyboard. In the second evaluation, our participants reached a speed of 108 cpm (characters per minute [21.6wpm]) and an error rate of 4% which is close to the performance of standard users on classic touchscreen keyboards. The third evaluation additionally revealed that our participants increased their typing speed with the Gestyboard by 44% and decreased their error rate by 48% in just 3 trial sessions. This steep learning curve is mostly due to the familiarity to the QWERTY layout.

Keywords: Gestyboard, Iterative UI Development, Text input, Gestures, ten-Finger-System, UI Evaluation, User-Centered Design

1 Introduction

Besides the well-known advantages of touchscreens there are also disadvantages compared to the classical WIMP-based interaction with mouse and keyboard. One important disadvantage is the lack of tactile feedback. Many studies have focused on solving this challenge by either introducing new touchscreen technologies that offer some kind of tactile or haptic feedback or by developing new, suitable touchscreen-based keyboard concepts. This paper presents the second iteration of a novel text input concept, called **Gestyboard**, designed for touchscreens. The name is derived from the fact that unique finger-based **gestures** (taps and slidings) are used to activate keystrokes. The intention of this core characteristic is to tightly link the Gestyboard text input mechanism to the

10-finger-system¹. For example if the user performs a tap gesture with the left pinky finger, the system will always interpret this as the letter 'a', immaterial of where the tap is executed. On sliding the pinky finger up, the letter 'q' will be typed. This way, it is impossible to miss-type the intended key or to accidentally hit an adjacent key while performing the gesture with the correct finger. However, errors still can occur by using the wrong finger or more than one finger simultaneously for a single intended gesture.

The goal of this concept is to enable high performance and blind typing without the need for tactile feedback. This can be achieved by abiding with well-known concepts like the QWERTY layout and the 10-finger-system.

2 Related Work

With mobile computing becoming more and more ubiquitous the development of new keyboard concepts have also rapidly evolved. In spite of this, the performance of touch-screen-based keyboard concepts still suffers, primarily due to the lack of tactile feedback [9]. Goldberg et al. [7] developed a text input concept called Unistrokes based on simple gestures. These gestures were designed to satisfy three major criteria: simplicity in learning, simplicity in execution and uniqueness. The execution of the gestures does not rely on tactile feedback, this makes blind typing possible. However, learning of those gestures is necessary. Another text input concept developed for mobile use is called Swype[14]. This concept is based on a dictionary and the QWERTY layout. Here, the user strokes through the letters of the desired word. By combining this approach with error-correction, words can be recognized correctly even if the swipe is not precise.

2.1 Stationary Text Input Concepts

Attempts have also been made to increase the performance of text input for larger touch-screen devices like multi-touch monitors or tabletops. In 2008, Patrick Bader developed a touchscreen keyboard that ergonomically adapts to the users hand [2]. Figure 1 shows the layout of this keyboard which is based on the QWERTY layout. The fingers rest on the home row keys and a keystroke is performed by tapping the keys.

One year later, in 2009, Microsoft published the US patent 2009/0237361 A1 which is a QWERTY keyboard split into two halves. On placing the hands on the touchscreen, the left half of the keyboard appears below the left hand while the right half below the right hand. Additionally, the two halves follow the hands of the user while moving or rotating. Sax et al. improved this concept by dividing the keyboard into ten parts – one for each finger[13]. Each part follows the corresponding finger and keystrokes are performed with taps.

¹This is also called touch typing in literature. However, to avoid confusion in the context of touchscreens, the term 10-finger-system will be used here.

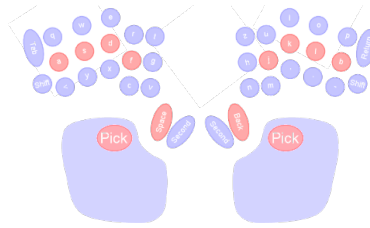


Fig. 1. The layout of the touchscreen keyboard from Patrick Bader

2.2 Text Entry Evaluation

Typing-speed is usually measured in words per minute (wpm). One word is defined as a sequence of 5 characters inclusive of spaces and signs [1] ($5cpm = 1wpm$). Another important parameter characterizing the performance of the text input is the error rate.

Apart from these performance oriented factors there are other equally important aspects to consider when comparing different text input concepts. These include the acceptance, the learnability, the accessibility, the scalability, and the size needed for the visual representation. Different approaches for evaluating text input methods have been presented in literature. One such method is to let the participants type freely any text that comes to their mind [10]. The advantage is that time is not spent in looking at the predefined text in order to follow it. However, using this method makes it impossible to differentiate between mistakes and intended words. It is also not possible to ensure that the distribution of the letters matches the natural distribution found in the corresponding language.

For those reasons, we chose to use another technique: The participants see the text they should type either on a sheet of paper or on a screen. Since each participant types the same text under these controlled circumstances, the individual results can also be better compared. We used the phrase set of MacKenzie and Soukoreff to ensure that the frequency of the letters are similar to their frequency in the English language[10]. To further ensure that each letter appears at least once in our text, a pangram was part of each subset.

3 Requirements

We believe that the classical keyboard can only be replaced by a text input mechanism if the same requirements are fulfilled that are already met by the classical keyboard. Based on this assumption, we collected the following requirements. These are also the requirements which form the foundation of the **Gestyboard** concept.

(R1) No additional hardware: Additional hardware slows down the distribution, and therefore decreases the acceptance of the concept.

- (R2) No dictionary:** While a language model can improve the performance of a text entry technique, it can also reduce the power of the technique.
- (R3) Comparable performance to the classic keyboard:** Any new concept needs to be at least as fast and robust as the classical input. Otherwise the new concept can not compete.
- (R4) Avoid missing the intended keys:** The main reason for the worse performance of a standard touchscreen keyboard is hitting accidentally the wrong keys because of the lack of tactile feedback.
- (R5) Enabling blind typing:** Blind typing increases the performance of the user because the attention can be focused on the text instead of searching for keys.

Table 3 shows a small survey of touchscreen-based text input concepts developed in the last decade and whether they fulfill R1 to R5 or not.

Concept	Year	R1	R2	R3	R4	R5
TCube [15]	1994	no	yes	no	no	no
Quikwriting [12]	1998	no	yes	no	no	no
Dasher [16]	2000	yes	yes	yes	no	no
EdgeWriting [17]	2003	no	yes	no	yes	yes
Graffiti [4]	2008	no	yes	no	no	yes
Bader keyboard [2]	2008	yes	yes	no	no	no
Microsoft Split [11]	2009	yes	yes	no	no	no
Fitaly [6]	2010	yes	yes	no	no	no
LiquidKeyboard [13]	2011	yes	yes	no	no	no
Swype [14]	2011	yes	no	no	no	no

Table 1. Several text input concepts and their relation to our requirements. In this table a requirement is fulfilled if we found an evidence for it.

4 Concept

The Gestyboard is activated when the user touches the screen with all ten fingers. A visual representation of the Gestyboard is then displayed showing the QWERTY layout as illustrated in Figure 2 a). The home row key letters 'a','s', 'd', 'f', 'j', 'k', 'l' and ';' appear directly below the fingers of the user (see Fig. 2 a). Each finger is associated with a group of keys, with the home row key as the center key (see Fig. 2 b).

Gestures

The Tap Gesture: To type one of the home row letters simply a tap with the appropriate finger is required. Each finger is only allowed to type letters according to the ten-finger-system. For instance, if the user wants to type an 'a', the left pinky finger is used. Even if the graphical representation of another letter is hit an 'a' appears. This feature eliminates the requirement of the user to physically look at the graphical representation

of the Gestyboard. This enables blind typing, hence meeting one of the most challenging of the requirements listed (R5, see section “Requirements”). Moreover, the group of keys which are linked to the pinky finger is re-centered at the position where the tap was performed.

The Sliding Gesture: The keys positioned directly below or above the home row keys are typed by sliding the appropriate finger down or up, respectively. However, in the case of the pinky and index fingers additional sliding gestures have to be defined, due to additional keys located horizontally and diagonally adjacent to those home row keys. For example if the user desires to execute the enter key, then the right pinky finger must be slided to the right.

An overview of all gestures for each finger is given in Figure 2 b).

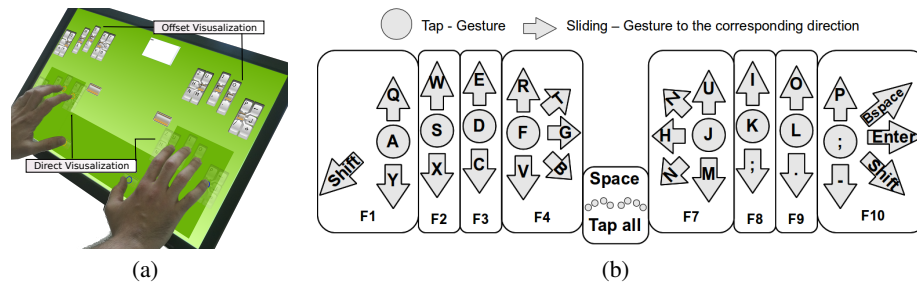


Fig. 2. a) The graphical representation of the Gestyboard b) The gesture overview. Fx = Finger x

Space activation gesture: To conform to the rules of the ten-finger-system, the Space key was activated by tapping with either the left or the right thumb in the first version of the Gestyboard. However, we discovered in our first evaluation that space keystrokes were executed accidentally. The reason for this was that due to their larger size, thumbs sometimes were recognized not as one, but two touch points. This was interpreted as a rapid alternation between two corresponding touch points by the software. To overcome this source of errors a new Space key activation metaphor was implemented. To input a space, a simultaneous tap with all ten fingers has to be performed. As a positive side effect, each time a space is typed, all keys are re-centered according to the positions of the user’s fingers.

Challenges

This section describes the most important challenges and the proposed solutions.

Move fingers independently: One basic challenge can be attributed to human anatomy. Without further training people are usually not able to move their fingers completely independent from each other. According to the feedback of our first test users, the separate movement of the ring fingers is even more challenging than that of the other fingers.

Hit wrong keys: Another result of the first evaluation was that users were accidentally hitting additional keys as they tried to perform the more challenging diagonal sliding gestures. The reason for this can be seen in Figure 3. On the left side the less error-prone approach is shown. A keystroke is only performed when the user re-centers the finger after the visual representation of the letter was reached. That way it is guaranteed that the finger of the user always returns to its appropriate home row letter. However, to support more advanced users the second alternative was implemented, which is illustrated on the right side of Figure 3. The user is still able to re-center the finger, but for some words there might be a gain of speed by following the second alternative. However, this was one of the main sources for errors in our first version of the Gestyboard.

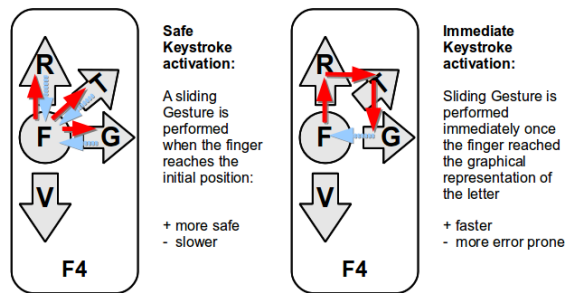


Fig. 3. Two alternatives to type the letters “RTG”. Left side: The finger always has to be re-centered to activate a keystroke. Right side: A keystroke is activated when the key is reached by the finger.

Learning Curve: One goal of the evaluation is to determine if our solutions for the mentioned challenges improved the Gestyboard’s performance. Another goal is to find out how fast people adapt to this input method over the course of multiple sessions. To be able to compare the performance of a new text input concept with the performance of the classic keyboard, the users need a lot of training. An evaluation over three sessions typing 1000 characters each was performed to gain an initial insight on the associated learning behavior.

5 Evaluation

The results of the first formal evaluation (Pilot Study) with 42 pupils trained in ten-finger touch-typing described in [5] showed a lot of enthusiasm for the concept among

the users. The second (Proof of Concept) study is described in the first part of this section while the third (long-term) evaluation is described in the second part of this section.

5.1 Second Evaluation: Proof of Concept

Participants Two developers having limited experience with the Gestyboard were chosen as the participants for this evaluation. Due to their experience in this field and their prior exposure to the concept they could provide valuable insights on how the technique could be improved.

Procedure In order to detect some learning effects, each participant had to complete eleven sessions. 1371 characters were typed in session eight and eleven while in each of the remaining sessions, 248 characters were typed. There was a time interval of one to two day between the sessions. For each session we chose a different subset from MacKenzie's phrase sets for evaluating text entry techniques [10].

5.2 Third Formal Evaluation: Learning Effect

We conducted yet another study involving unbiased test subjects. The principles of the third formal evaluation are described in the following:

Participants: We evaluated the system with 12 students (11 male 1 female). All were between the ages of 21 and 32 years and have had no prior experience with the Gestyboard. Moreover, none of the participants were accustomed to the ten-finger-system.

Procedure: The evaluation was a within-subject design and consisted of three individual sessions, conducted on separate days with a time period of two to three days. For each session a different subset of MacKenzie's phrase sets [10] was used. The participants were asked to enter a text with 1000 characters per session. In the first session our participants typed the phrase set using the Gestyboard, the classical touchscreen keyboard and the classical hardware keyboard (3 stations). There was a ten minute break granted between each station. During this break the participants filled out the System Usability Scale (SUS) questionnaire [3] for each station. The second and third session had exactly the same setup except the classical hardware keyboard was excluded, as it was not necessary to collect learning effects on the already well known classical hardware keyboard. After the third session, the participants were interviewed to get qualitative feedback. On an average it took the participants 70, 52, and 47 minutes to complete the first, second, and third sessions, respectively.

6 Results

This section presents the results of the conducted evaluations.

6.1 Second Formal Evaluation: Proof of Concept Results

The two participants of the proof of concept study were familiar with the Gestyboard. As expected, the speed increased while the error rate decreased throughout the sessions. Figure 4 a) shows the results of the two participants individually. The measured typing speed was 108 cpm (21.6 wpm) and the lowest error rate was 4% in average.

6.2 Third Formal Evaluation: Learning Effect Results

Learning Effect: Figure 4 b) shows the cpm, the number of errors and the time needed to type 1000 characters for each session. Average speed increased from session one to session three from 42 to 63 which is an increase of 44%. Simultaneously, the error rate decreased from session one to session three from 25 to 14 errors (-48%).

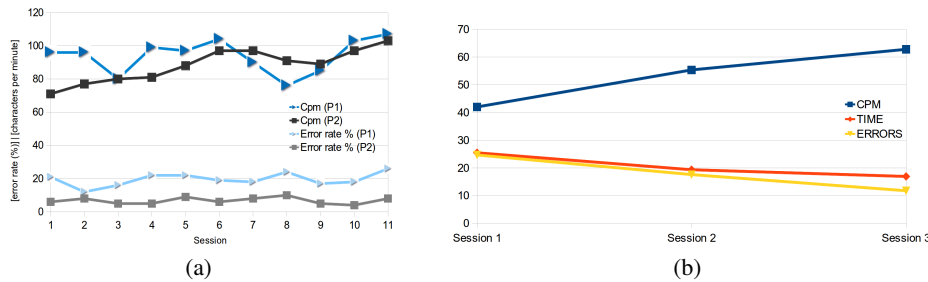


Fig. 4. a) Proof of Concept: Performance of two developers during eleven sessions with 276 characters per session. b) Mean values for cpm, errors and the time needed to type 1000 characters for each session.

Comparison to the classic touchscreen and hardware keyboard: Figure 5 shows the comparative results of the characters per minute and the absolute error value for each participant. Overall, the Gestyboard still does not perform as good as the classic virtual touchscreen keyboard and the classic hardware keyboard (both in speed and error rate).

Comparison between Gestyboard 1.0 and Gestyboard 2.0 Figure 6 compares the first version of the Gestyboard with the current version. While the average cpm was 31 for the previous version of the Gestyboard, it increased to 45 for the current version, which is an improvement of 45%. The number of errors decreased from 42 to 24, which is a decrease of 42%.

Figure 7 shows the mean error rate in percentage for each character of Gestyboard 1.0 and Figure 8 shows the same for the current version of the Gestyboard. By comparing these two figures it is noticeable that the number of errors made for each character on version 2.0 of the Gestyboard diminished significantly compared to its predecessor.

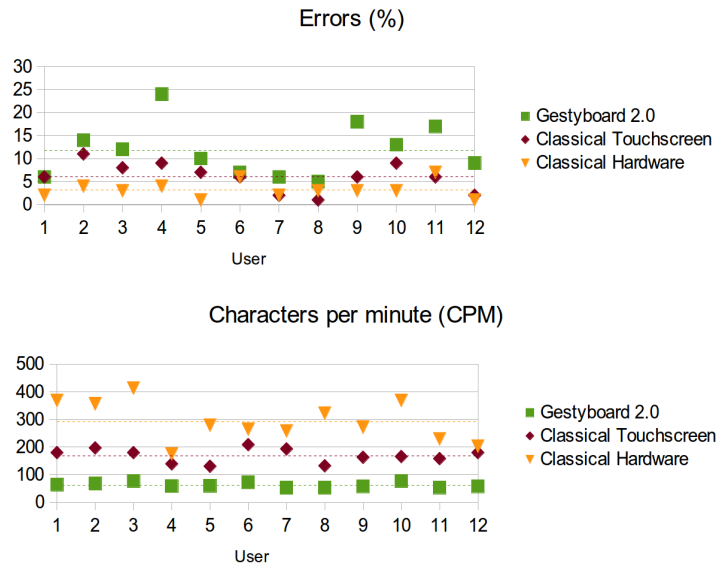


Fig. 5. Comparison between the Gestyboard, the classical touchscreen keyboard and the classical hardware keyboard for each user (Last session). The lines represents the mean values.

This is especially the case for the letters 'x', 'b', 't', 'n', and 'k'.

7 Conclusion & Discussion

In the second formal evaluation the participants reached a typing speed of 108 cpm (21.6 wpm). According to Sax et al. [13] this is already a competitive result when compared to the classical virtual touch-screen keyboards. Sax et al. argued through the work of Lopez et al. [8] that the mean typing speed of classical touchscreen keyboards is 92.5 cpm (18.5 wpm). However, our evaluation of the classical touchscreen based keyboard resulted in a mean typing speed (in our third session) of 168 cpm (33.6 wpm). This difference to the results given in [13] can be attributed mainly to two factors: First, our participants also improved their typing speed for the classical touchscreen keyboard during our sessions. Second, and even more important, the evaluation in [8] was performed on an Apple iPhone which is a small screen device and therefore not as ergonomic as a keyboard used on a large screen. Consequently, the results of our second formal evaluation prove that the Gestyboard can compete with touchscreens on small screen devices but not yet on standard touch-screen keyboards.

The participants in our third formal evaluation could not yet reach the performance of the classic touchscreen keyboard (62.8 cpm and 11.75% error rate), but it nevertheless revealed that there is a fast learning behavior. Our participants improved their typing

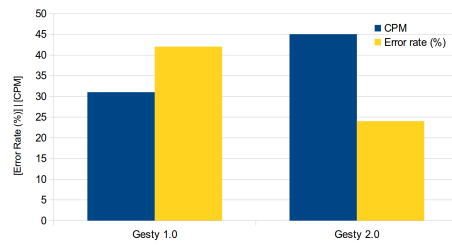


Fig. 6. Comparison between Gestyboard 1.0 and Gestyboard 2.0

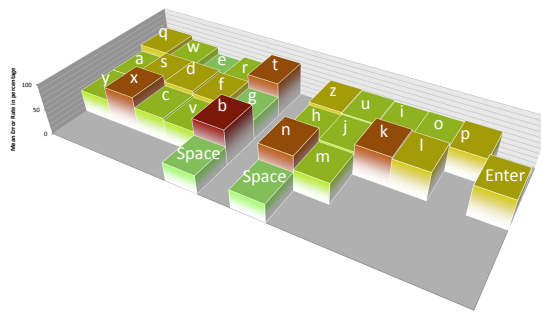


Fig. 7. Errors per character for the first version of the Gestyboard (1.0)

speed by 44% with an average of 63 cpm in just three sessions, and reducing their errors by 48%. The performance can increase dramatically once a user familiarizes with the keyboard layout and the movements on the Gestyboard. A potential benefit from the ten-finger touch-typing method is also evident. Additionally, some of our participants reported that they also became habituated to the finger movements required for a whole sequence of gestures. This was more apparent for words which they typed frequently. This not only improves the typing speed, it also enables the user to blind type without any haptic feedback.

Additionally, the participants reported errors made due to moving of the wrong finger. A large amount of time being spent in search of the correct letters. However, this was more the case at the beginning of each session, where the participants required time to re-acquaint themselves with the gestures.

The space key gesture was another source of error. When users remove their hands for a short period of time and then position their hands back on the screen, the system detects the entry of a space key. However, a space key was often initiated even when the users just wanted to relax their fingers or recenter them. It also occurred when they wanted to abort an action.

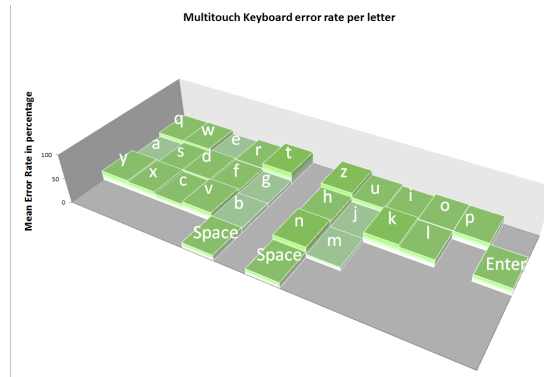


Fig. 8. Errors per character for the current version of the Gestyboard (2.0)

8 Future Work

Our qualitative interviews with our participants revealed some flaws in the current implementation and the concept itself. To remove one such flaw, we decided to change the gesture for the space key from tapping with all ten fingers to merely tapping with the five fingers of the left hand. The same gesture performed with the right hand would then be used for the backspace key. This way, people can remove both hands without the risk of performing the space gesture.

We are also investigating algorithms to keep the group of keys following the individual fingers when the movement of the finger is not intended by the user, as fingers tend to drift on the screen over time. For this, the speed of the finger might be an indicator.

To eliminate another source of errors, the tap gesture will be optimized. To enable the user to cancel a tap, the tap gesture will be aborted when the user immediately slides the finger a certain amount after the finger touches the screen again.

By eliminating those weak points, the performance of the Gestyboard itself can be further improved.

In addition to these improvements, we plan to conduct an extended long term evaluation. This way, the performance of the Gestyboard can be better compared with the performance of the classical keyboard. We will then choose those participants who claim to be already proficient in blind typing to conduct another evaluation in which the visualization of the Gestyboard is hidden. We will do the same with the classical touchscreen keyboard (providing some visual or haptic feedback to know where the keyboard is) and compare the results with each other.

References

1. Arif, A., Stuerzlinger, W.: Analysis of text entry performance metrics. In: Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference. pp. 100–105. IEEE (2009)
2. Bader, P.: Entwicklung eines virtuellen Eingabegeräts für Multitouch Screens. Bachelor's thesis, Hochschule der Medien, Stuttgart, Germany, GE (Aug 2008)
3. Brooke, J.: SUS: a 'quick and dirty' usability scale. Usability evaluation in industry (1996)
4. Castellucci, S., MacKenzie, I.: Graffiti vs. unistrokes: an empirical comparison. In: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. pp. 305–308. ACM (2008)
5. Coskun, T., Artinger, E., Pirritano, L., Korhammer, D., Benzina, A., Grill, C., Dippon, A., Klinker, G.: Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile. techreport (2011), <http://drehscheibe.in.tum.de/forschung/pub/reports/2011/TUM-I1122.pdf.gz>
6. Fitaly: *Fitaly*. www.fitaly.com/fitaly/fitaly.htm, 02.04.2012
7. Goldberg, D., Richardson, C.: Touch-typing with a stylus. In: Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems. pp. 80–87. ACM (1993)
8. Lopez, M., Castelluci, S., MacKenzie, I.: Text entry with the apple iphone and the nintendo wii (2009)
9. Luk, J., Pasquero, J., Little, S., MacLean, K., Levesque, V., Hayward, V.: A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. pp. 171–180. ACM (2006)
10. MacKenzie, I.S., Soukoreff, R.W.: Phrase sets for evaluating text entry techniques. In: CHI '03 extended abstracts on Human factors in computing systems. pp. 754–755. CHI EA '03, ACM, New York, NY, USA (2003), <http://doi.acm.org/10.1145/765891.765971>
11. Mosby, T.J., Wiswell, C.N.: Virtual keyboard based activation and dismissal. US-Patent 2009/0237361 (sep 2009)
12. 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)
13. Sax, C., Lau, H., Lawrence, E.: Liquidkeyboard: An ergonomic, adaptive qwerty keyboard for touchscreens and surfaces. In: ICDS 2011, The Fifth International Conference on Digital Society. pp. 117–122 (2011)
14. Swype: *Swype*. www.swypeinc.com, 02.04.2012
15. Venolia, D., Neiberg, F.: T-cube: a fast, self-disclosing pen-based alphabet. In: Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence. pp. 265–270. ACM (1994)
16. Ward, D., Blackwell, A., MacKay, D.: Dasher data entry interface using continuous gestures and language models. In: Proceedings of the 13th annual ACM symposium on User interface software and technology. pp. 129–137. ACM (2000)
17. Wobbrock, J., Myers, B., Kembel, J.: Edgewise: 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. pp. 61–70. ACM (2003)