Gestyboard BackTouch 1.0: Two-Handed Backside Blind-Typing on Mobile Touch-Sensitive Surfaces

Tayfur Coskun(✉), Christoph Bruns, Amal Benzina, Manuel Huber, Patrick Maier, Marcus Tönnis, and Gudrun Klinker

Technische Universität München, Boltzmannstr. 3, 85748 Munich, Germany
coskun@in.tum.de
http://ar.in.tum.de/Main/TayfurCoskun

Abstract. This paper presents a new and innovative gesture-based text-input concept designed for high-performance blind-typing on mobile devices with a touch-sensitive surface on their back-side. This concept is based on the Gestyboard concept which has been developed by the Technische Universität München for stationary use on larger multi-touch devices like tabletop surfaces. Our new mobile concept enables the user to type text on a tablet device while holding it in both hands, such as the thumbs are in the front of the tablet and the other eight fingers are in the back. The user can hence type text using these fingers on the back of the device. Although, the gesture-based finger movements are quite unfamiliar and the participants need to mentally rotate the QWERTY layout by −90 and 90 degrees respectively, our multi-session evaluation shows that despite the fact that their fingers are occluded by the tablet, our concept enables the users to blind-type and that they improve their performance in each session. Consequently, the user can use all ten fingers simultaneously to type text on a mobile touchscreen device while holding it comfortably in both hands. This implies that our concept has a high potential to yield to an high-performance text-input concept for mobile devices in the near future.

Keywords: User interfaces · Mobile blind typing · 10-finger-system · Touch-typing · Gesture · Touchscreen

1 Introduction

The fast and the wide spreading of tablet PCs in the recent years increased the need for high performance touchscreen-based text-input techniques. While a comfortable and ergonomic way to interact with a tablet is to hold the tablet PC in both hands while interacting with it, the user however is restricted to the usage of the thumbs which reduces the efficiency. Even if the keyboard is split in a way that all keys can be reached with the thumbs (e.g. a virtual split keyboard), the user's typing performance suffers from the latter restriction.

© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2014
I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 422–434, 2014.
DOI: 10.1007/978-3-319-11569-6_33

The contribution of this work is to enable text-input for the eight free fingers on the back of a mobile device by positioning virtual invisible keys in an user-adaptive ergonomic manner on the backside of the tablet, rather than optimizing virtual keyboards with dictionaries (e.g. Swype [14]) or optimized layouts (e.g. 1Line keyboard [8]). Of course, using dictionaries for text-input improves the performance a lot. But our goal is to eliminate the real source for the lower text-input performance on touchscreens when compared to the classical hardware keyboard. The source for the lower text-input performance on touchscreens is the missing tactile feedback and hence the missing possibility to blind-type. Additionally, when holding the tablet in both hands, the user is restricted to the two thumbs to type text. Since, like the original Gestyboard concept, our new mobile concept is based on unique finger-based gestures for each group of keys defined by the 10-Finger-System, there is simply no need to observe the own fingers while typing, once the gestures are learned by heart or even better by the muscle-memory of the hands of the user. In the latter case, the user stops thinking about the movements and the hands are just performing what the user wants to type like it is the case for expert users on the classical hardware keyboard. The Gestyboard concept is described in more detail in Sect. 3. Our new concept for mobile use is described in Sect. 4.

2 Related Work

Providing a high performance touch-screen based mobile text-input system is a special challenge since the first smartphones started spreading out some years ago. Since then, the quality of the touchscreen itself (sensitivity, accuracy) as well as the developed text-input concepts improved a lot. But still, compared to the classical hardware keyboard, the user is very limited in the case of touchscreen based text-input concepts. One reason behind is that usually tactile feedback is missing on touchscreen. Hence, the user has to put more mental focus on the keyboard to be able to hit the correct keys with a naive button-based implementation of the touchscreen keyboard. Another reason is the limitation to one or two thumbs when the tablet is hold in one or two hands respectively or the limitation to a few fingers on one hand, while the tablet is hold in the other hand. This limitation of course has an negative impact on both, the speed and the accuracy of typing text. While the user can use all 10 fingers on the classical keyboard, it is very difficult and most of the time impossible to use all ten fingers to type. Some keyboard concepts focus on eliminating the first mentioned limitation, the missing tactile feedback by providing a hardware solution for providing some kind of haptic feedback in general. Since the accuracy of hitting the right button is naturally worse when compared to the classical keyboard, the currently best working solution is, instead of expecting the user to hit the correct key, to enhance the keyboard concept with a dictionary. Thus, the most successful keyboards on nowadays smartphones and/or tablets guess what the user wanted to type instead of what actually was typed. One example is called SwiftKey [13] and started to be developed in 2008. Swiftkey analyses personnel text of the user (E-Mails, SMS,...) to learn the style of typing of the specific user. This way, SwiftKey does not only guess what the user wanted to type it also predicts the next word the user intends to type by using the knowledge it learned from the user's text messages. Although it is very difficult to find some information about the average WPM of commercial keyboard concepts like the SwiftKey, we know from personnel reports that the users feel quite fast and successful when using it. Another famous example, also using a dictionary developed in 2011 is called Swype [14]. Instead of hitting buttons representing the keys the user performs a sliding gesture on the keyboard including all letters of the word the user wants to type. Swype then searches for all possible word combinations and suggests the correct word with a high probability. Both concepts even has been imported from some smartphone manufactures like Samsung or HTC and provide similar techniques on their own soft-keyboards. Although, the performance and the user acceptance of the mentioned keyboards is good and they also make sense for the use with smartphones, they are still not replacing the classical hardware keyboard. One reason for that is, that the users are used to the classical hardware keyboard for decades. Another reason is that the classical keyboard allows the user to type anything, even a sequence of letters which doesn't seem to make any sense and therefore does not exist in and dictionary. This is for example the case for typing passwords, writing code or just to chat in the user's own personalized way of writing like chatting or writing an E-Mail to a close friend or changing the language frequently.

For those reasons, we decided to additionally solve the second mentioned limitation of touchscreen-based text-input concept in the case of tablets, which is the limitation to one or both thumbs when holding a tablet in both hands (or to a few fingers when hold in one hand) and to enable the user to blind-type. It is well-known that a comfortable way of interacting with a tablet is to hold it in both hands and use the two thumbs to interact with the device. Microsoft provides for example a split keyboard on Windows 8 which can be reached with the thumbs. They use a previous study, to determine the positions of the keys of the splitted keyboard which was conducted in 2012 [11]. Swiftkey also provides on option to split the keyboard in two parts. While this way of interacting with the tablet is very comfortable, 8 fingers are wasted by just holding the tablet and the least precise fingers, the thumbs, are used to interact with the tablet. This of course makes sense, since first nowadays tablets usually does not provide a touch sensitive surface on the back side of the device and second the tablet occludes the fingers of the user, hence making it difficult to use them to interact with the tablet. However, devices with touch sensitive surfaces on their back indeed exists already (e.g. Playstation Vita [12]) and we expect, that this technology will be more distributed in future. Another example for enabling touch on the back of very small devices has been published in 2009 from Baudisch et al. [2]. For this, small prototype devices were built and a cursor is used to visualize the position of the users finger, hence enabling the user to interact with UI elements on the front side of the device. This way, Baudisch et al. also provides a technique solving the second problem, the occlusion of the fingers.

Wigdor et al. [16] investigated backside interaction features with a pseudo transparent display which is a similar approach but for tablets visualizing the whole fingers of the users, not only it's position with a cursor. The fingers on the back of a tablet device were shown as half-transparent shadows to ease input mapping. In their studies, besides other touch controls, Wigdor et al. implemented a simple button based keyboard. The keyboard was either split like conventional keyboards, or split and reoriented to better fit to the orientation of the hands. The participants were able to hit the buttons behind the see-through tablet. In our Back-Type solution, we also use cursors to visualize the finger's position on an external screen and also reorient the rotation of the keyboard. However, well trained users should be able to interact with our text-input concept, even without any visualization in future. For this, the Blind-Typing text-input concept for large touchscreens like tabletop called Gestyboard [4] can be adapted. Kim et al. [6] installed a physical keyboard on the back side of a smartphone with small buttons representing the QWERTY keyboard and conducted an evaluation. An average speed of 15.3 WPM were with an error rate of 12.2 %. Also in 2012 Wolf et al. [17] conducted a study in which the effect of gestures on the back of a tablet device were examined. For this, they attached two IPads to each other for evaluation purposes.

Since our mobile concept uses the original Gestyboard concept to provide unique finger gestures, it is described in further detail in Sect. 3. Afterwards, in Sect. 4, our mobile and adapted version of the Gestyboard concept and the difference to the original Gestyboard are explained.

3 The Original Gestyboard Concept for Stationary Use

This work adapts the Gestyboard concept which has been developed and evaluated by the Technische Universität München [4] in 2011. The 10-finger-system[1] and the QWERTY layout are used to define individual gestures for each finger. This way, each finger is only able to type the set of letters defined by the 10-finger-system. For convenience, especially for the people not mastering the 10-finger-system, a visual representation of the set of letters was displayed during the evaluation below each associated finger. Figure 1 shows the original representation of the Gestyboard. It is important to note that the visual representation of the keys actually is not the keyboard itself. All the keys are activated through unique finger gestures instead of hitting a button. In fact, for the final usage of the Gestyboard it is planned to hide the keyboard completely. It is automatically activated by touching the screen with 10 fingers and the user can just start typing on the interface of any application.

Key activation. The finger gestures can be either a tap or a sliding. To activate the home row key letters ('a', 's', 'd', 'f', and 'j', 'k', 'l', ';'), the user performs a tap gesture with the associated finger. This means, even though if the user

[1] This is also called "touch typing" in literature. However, to avoid confusion due to the association with the touchscreen, we use the wording "10-finger-system" instead.

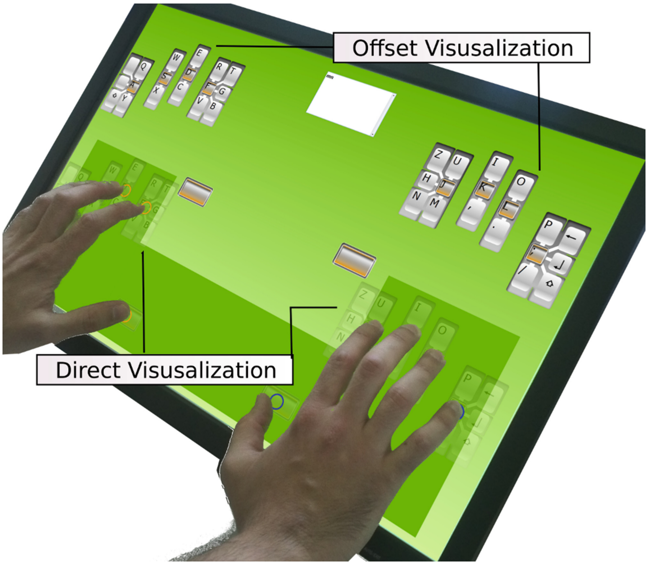

Fig. 1. The Gestyboard's visualisation

accidentally hits, for example, the visual representation of the letter 's' tapping the left pinky finger, the system will still correctly type the expected letter 'a'. In fact, tapping with the left pinky finger is a unique gesture associated to the letter 'a'. Consequently, the user can close the eyes and type an 'a' easily without worrying about the exact position of its visual representation. To activate the remaining neighboring letters, a sliding gesture has to be performed with the associated finger into the associated direction. Finally, to type a "space", all fingers have to simultaneously perform a tap gesture. This is actually the only difference to the 10-finger-system. Hence leading to an automatic reset of all fingers positions between the words. Figure 2 gives an overview about all gestures in the 2.0 version of the original Gestyboard concept. The space activation gesture has been changed to perform a tap gesture with all ten fingers. The initial reason behind was the low tracking precision of the touchscreen when the thumbs were touching the screen. The touch point detected by the system was alternating between two different positions below the thumbs which leaded to multiple touch-up and touch-down events. Our system interpretes this as a tap with the thumb which leads to wrong a space keystroke. The hand gesture were accepted by the users very well, because of that we kept this gesture for the backtouch version of the Gestyboard. As a positive side effect, the thumbs are free for any interaction with the active interface of the tablet device.

The scope of the original concept. The original Gestyboard concept is designed to be used on a large multi-touch device to replace the classical hardware keyboard. This goal could not be reached yet according to the authors, but it could be shown that the idea of building a touch-based text-input concept based on the

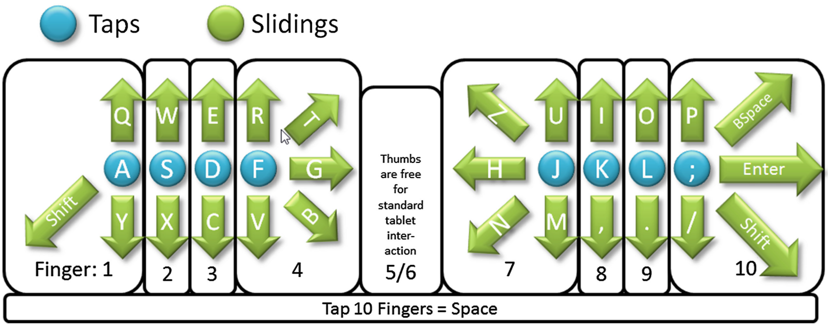

Fig. 2. The Gestyboard's gestures overview in version 2.0. Fx = Finger x.

10-finger-system works [5] and enables experienced users to blind-type. Since this concept benefits from the experience with the 10-finger-system, people lacking this experience are quite slow and error prone when using it the first time. However, most of the 42 participants of the first evaluation of the Gestyboard concept liked it and were motivated to learn it. Due to this fact, a second study followed consisting of 6 sessions with 1000 characters each. The results showed that people could reach more than 20 wpm (words per minute) with less than 2 % error rate. Encouraged by those promising results of the second iteration of the Gestyboard, our adapted concept for mobile use tackle the main challenge of allowing the user to blind type on the back-side touch surface while the fingers are not visible. As mentioned earlier, the goal of this paper is to investigate, if the Gestyboard concept can be adapted in a way people can use it on the back side of the device and to find out how they perform in multiple test sessions.

4 Gestyboard BackTouch: Transferring the Gestyboard Concept to the Back of a Mobile Device

As stated earlier, the Gestyboard concept enables the user to blind-type on a touchscreen. This concept can therefore be transformed and adapted to blind-type with the free 8 fingers on the back-side of a tablet device while holding it in both hands. The direct effect of this transfer on the users is the need to mentally rotate by $-90°$ the left half side of the keyboard and by $90°$ the right half side. Additionally, both sides of the keyboard have to be vertically mirrored (see Fig. 3). This means, that the keyboard is transformed the same way as the hands of the user. Consequently, the keyboard layout remains aligned to the fingers' orientation.

5 First Gestyboard BackTouch Prototype

Compared to the original Gestyboard, the finger movements of our adapted version are apparently more challenging. This emerges from the modification of

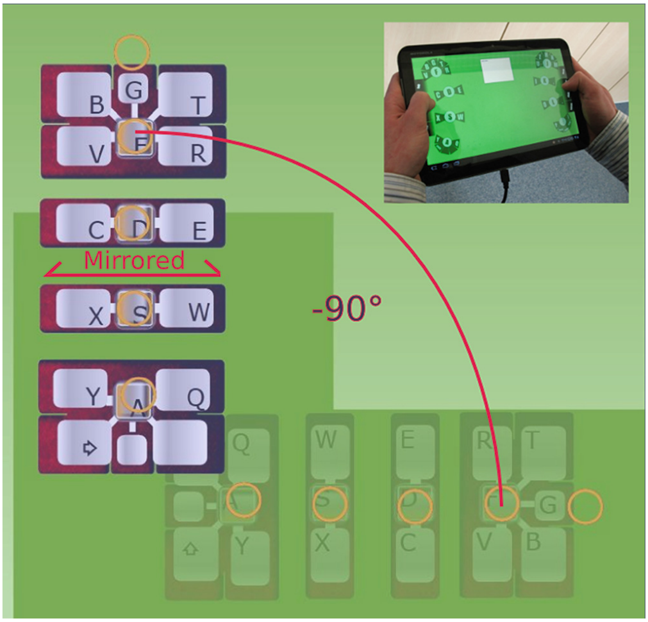

Fig. 3. The visual and mental adaptation of the Gestyboard concept and the resulting concept.

the QWERTY layout as shown in Fig. 3. Our first goal then is to see whether people are able to accommodate to the adapted concept. To be able to answer this question, an Android application has been developed for the Asus Transformer pad [1]. For evaluation purpose, the tablet is hold in a way that the front side of the device, i.e. the display, is on the back and hence is not anymore facing the holder. This way, the touchscreen of the device can be used to track the fingers' movements. For the final usage of course, a tablet with a touch sensitive surface on the back should be used. The gesture parameters are sent to the original Gestyboard application. Instead of reacting on the input of a directly attached touchscreen, our adapted version reacts on the data received from the network. This way, the Android tablet becomes a remote controller for the Gestyboard. For the final usage, the Gestyboard logic should be transferred to the tablet of course to have real mobile flexibility. Additionally, we added the capability to rotate and mirror the visualization of the key groups like described in Sect. 4. This way, the users see the visualization of their fingers' movements and the activated letters on an external monitor with the server running on it. This setup is also used during our evaluation and can be seen in Fig. 4.

This setup is a testing prototype. The future vision is to use a tablet with a touchscreen on the front side and a touch input system (e.g. a multitouch touchpad) on the back side. Additionally, it is planned to provide an option to

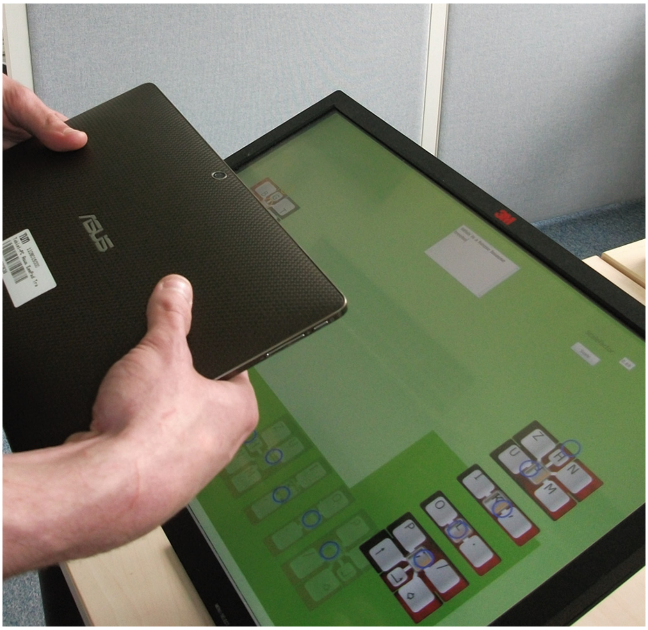

Fig. 4. Using an Android tablet as a remote controller for the original Gestyboard Prototype. The tablet is rotated to be able to test the concept.

disable the visualization of the finger movements and the gestures completely for expert users.

6 Evaluation

In this section we first describe the evaluation procedure and the participants in Sect. 6.1. The results are then presented in Sect. 6.2.

6.1 Procedure and Participants

It was planned to perform three evaluation sessions with ten computer scientist students for this initial test. After performing those three sessions with ten students, four among those ten asked us if they may continue with the evaluation to further improve their performance. Consequently, we decided to add another three sessions for those four participants.

The task of each session was to type 1.000 letters chosen from MacKenzie's phrase sets for text input evaluation purposes [10]. Those sentences all represent the letter frequency of real English sentences. Tipp10 software tool [15] was used to present and analyze the input data. This tool automatically collects useful data representing the user's performance in general and for each finger specifically and stores them in a database for further analysis. Finally, we asked

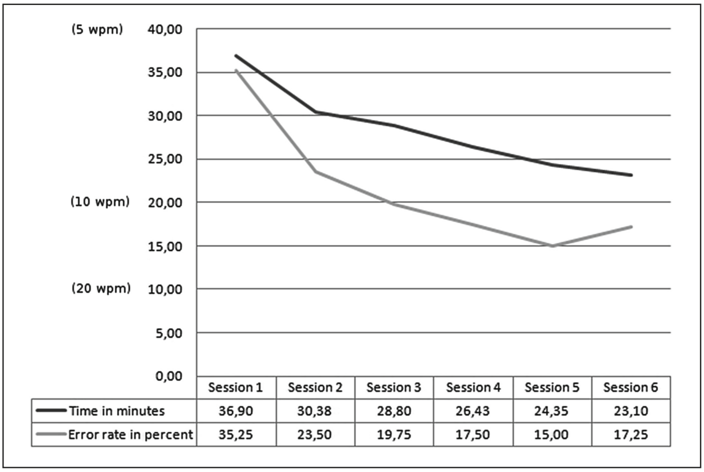

Fig. 5. Average time per session (minutes), error rate (percent), and the average speed (WPM)

the participants to fill out the System Usability Scale (SUS) [3] questionnaire and conducted a short interview.

6.2 Results

This section presents the quantitative evaluation results. The results are then discussed and interpreted in Sect. 7.

Performance. We obtained from Tipp10 tool the following quality measures: Typing speed, overall error rate, and the error rate per Finger. Figure 5 shows the average time needed to type 1000 letters per session in minutes (dark color curve) and the percentage of the error rate (light color curve). The Words per Minute (WPM) represents the average speed. The average speed in the first session among our 10 participants is 5.4 WPM while the error rate is 35.26 %. However, the speed is gradually increasing and the error rate is decreasing throughout the sessions. In the last session, the 4 remaining participants reached an increased average speed of 8.6 WPM and a reduced error rate of 17.52 %. Thus, despite the fact, that the tablet is occluding the fingers of the user and although there is no haptic feedback at all, our test users were able to blind-type with the 8 Fingers behind the tablet.

Error Rate per Finger. Figure 6 is showing the error rate in percentage per finger. The lowest error rates were reached with the home row keys (tap gesture) and the keys which are directly above or below them (up and down sliding gestures). The highest error rates were made with the keys placed diagonally to the home row keys (diagonal sliding gestures).

System Usability Scale. In this section we introduce the SUS values gathered from the SUS questionnaires for each participant which were filled out after

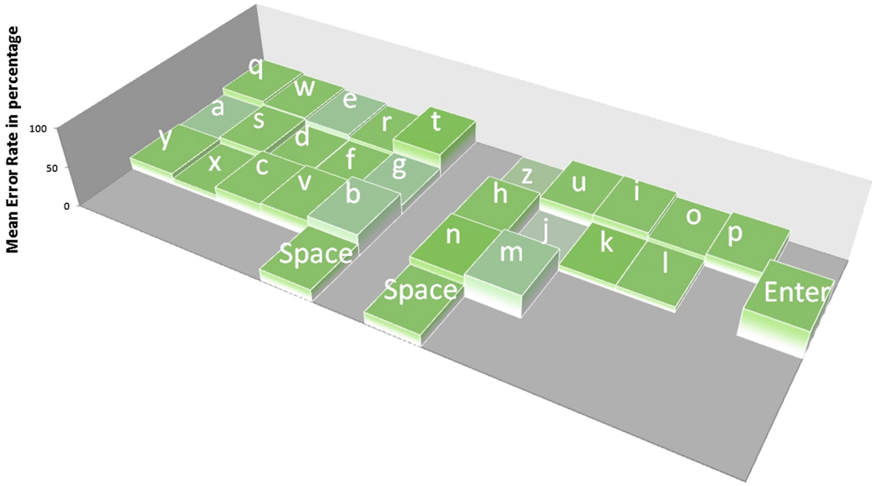

Fig. 6. Error rate per finger in percent.

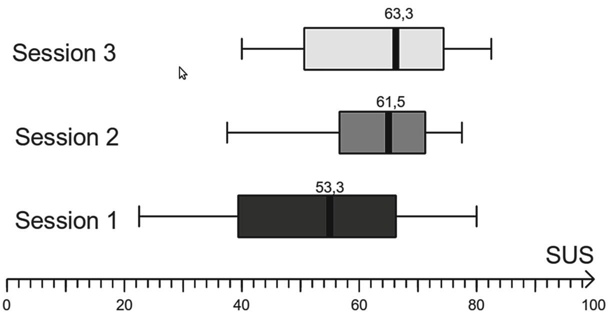

Fig. 7. System Usability Scale for each participant.

each of the three sessions to learn about the subjective initial usability rating of the users. SUS score is represented by a value ranging from 0 to 100, where 100 is the best score. Figure 7 shows the SUS scores for the first three sessions. We can clearly observe an overall increase in the SUS mean score throughout the sessions. Indeed, SUS score increased from a mean score of 53.5 in the first session to a mean score of 61.5 in the second one to reach finally a score of 63.3 in the third session. We also notice in the SUS boxplot that the distribution of SUS scores among users is getting narrower throughout the sessions and the maximum SUS score exceeded a value of 80 in the third session.

7 Discussion

From Sect. 6.2 we can observe two main conclusions. First, we notice that despite the observed increase, SUS score is still quite average, which makes it difficult to affirm for sure the users' feedback concerning the usability. However, we can conclude that the more training the users get, the higher the usability score is, which is somehow expected. This could also be interpreted from the narrower distribution in the SUS boxplot in the last two sessions. And second, we notice a clear increase in the learning curve of the Gestyboard BackTouch throughout the 6 sessions. However, 6 sessions with 255 letters in each session are not sufficient to compare these results with the classical hardware or touchscreen keyboards due to the large familiarity with the latter ones. MacKenzie et al. [9] for example developed an soft keyboard with an optimized soft keyboard and compared it with the qwerty layout on touchscreens. 20 sessions were conducted and the cross-over point between the performance of both layouts were reached after 10 more intensive sessions. In fact, the Gestyboard BackTouch (8.6 WPM) is not yet reaching the performance of text input system using the thumbs in edge interaction on a tablet PC (11 WPM) [7]. And this difference in the typing speed can be argued by the lack of experience with both the finger based gestures and the 10-finger-system. Therefore, with a better training, we expect the performance of the text input system using the thumbs to be surpassed by our solution, and the limitation to the thumbs interaction to be eliminated. Thus, although the performance in 6 quick sessions could not reach the performance of the well-established touch screen keyboards yet, we could prove that the Gestyboard stationary concept can be transferred to the back of the tablet device and that the user are able to adapt themselves to the innovative 10(or 08)-Finger-Based gestures. To reach the maximum speed possible, we expect our users need approximately a training of around 10.000 letters. This way, whole word gesture-sequences can be learned by muscle-memory, which will increase the performance of the Gestyboard BackTouch substantially.

8 Future Work

We will also correct some shortcomings of the implementation observed during the evaluation, like decreasing the load of exchanged finger gestures' events, to enhance our solution. Additionally, a long term and blind type evaluation will be performed in future. The next step is to provide a very simple kind of haptic feedback to the back of the tablet. For this, the tablet will be enhanced with a protection foil for touchscreens and the "'path"' of the finger gestures will be cut-out from this foil, hence providing haptic feedback. Haptic feedback showed to be very useful and efficient in improving the typing speed performance. If this simple kind of haptic feedback works, an ergonomically optimized hardware solution for touch-input on the back side of a tablet device would make sense and has the potential to become a high-performance and comfortable way of typing text on a tablet device without the need to occlude half of the limited tablet screen with a virtual keyboard. Additionally, it is planned to compare the performance of the Gestyboard BackTouch on different Android devices. The reason behind is that there was a noticeable difference in terms of accuracy when we tried to switch from the first ASUS Transformer to its next version, the

ASUS Transformer Prime. Unexpectedly, it was more difficult to use our concept on the newer device then it has been on the previous one, which has been used in our evaluation. It seems like that the new device was too thin and too light to provide the same grip as the previous one.

References

1. Asus. Asus transformer pad tf300t, 02 August 2012. http://www.asus.de/Tablet/TransformerPad/
2. Baudisch, P., Chu, G.: Back-of-device interaction allows creating very small touch devices. In: Proceedings of International Conference on Human Factors in Computing Systems, pp. 1923–1932. ACM (2009)
3. Brooke, J.: SUS: a 'quick and dirty' usability scale. Usability evaluation in industry (1996)
4. 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, Technische Universitaet Muenchen (2011)
5. Coskun, T., Artinger, E., Pirritano, L., Korhammer, D., Benzina, A., Grill, C., Klinker, G.: Gestyboard: a 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback (poster). In: SIGCHI - APCHI 2012 (2012)
6. Kim, H., Row, Y.-K., Lee, G.: Back keyboard: a physical keyboard on backside of mobile phone using qwerty. In: CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12, pp. 1583–1588. ACM, New York (2012)
7. Korhammer, D.: Development of a single touch user interface for the efficient and intuitive completion of forms in the area of emergency rescue services. Master's thesis, Technische Universitt Mnchen, Munich, Germany, GE, October 2010
8. Li, F., Guy, R., Yatani, K., Truong, K.: The 1line keyboard: a qwerty layout in a single line. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 461–470. ACM (2011)
9. MacKenzie, I., Zhang, S.: The design and evaluation of a high-performance soft keyboard. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the Limit, pp. 25–31. ACM (1999)
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, CHI EA '03, pp. 754–755. ACM New York (2003)
11. Odell, D., Chandrasekaran, V.: Enabling comfortable thumb interaction in tablet computers: a windows 8 case study. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 56, pp. 1907–1911. SAGE Publications (2012)
12. Playstation. Playstation Vita, 02 May 2012. http://de.playstation.com/psvita/
13. Swiftkey. Swiftkey, 11 June 2013. http://www.swiftkey.net/en/
14. Swype. Swype, 02 April 2012. http://www.swypeinc.com
15. Tipp 10. Tipp 10, 02 April 2012. http://www.tipp10.com/de/

16. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., Shen, C.: Lucid touch: a see-through mobile device. In: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, pp. 269–278. ACM (2007)
17. Wolf, K., Müller-Tomfelde, C., Cheng, K., Wechsung, I.: Pinchpad: performance of touch-based gestures while grasping devices. In: Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, TEI '12, pp. 103–110. ACM, New York (2012)