Sandwich Keyboard: Fast Ten-Finger Typing on a Mobile Device with Adaptive Touch Sensing on the Back Side

Oliver Schoenleben
Helsinki Institute for Information Technology
HIIT, Aalto University and University of Helsinki
Otaniementie 19B, 02150 Espoo, Finland

Antti Oulasvirta
Max Planck Institute for Informatics
Saarland University, Campus E1 7,
66123 Saarbrücken, Germany

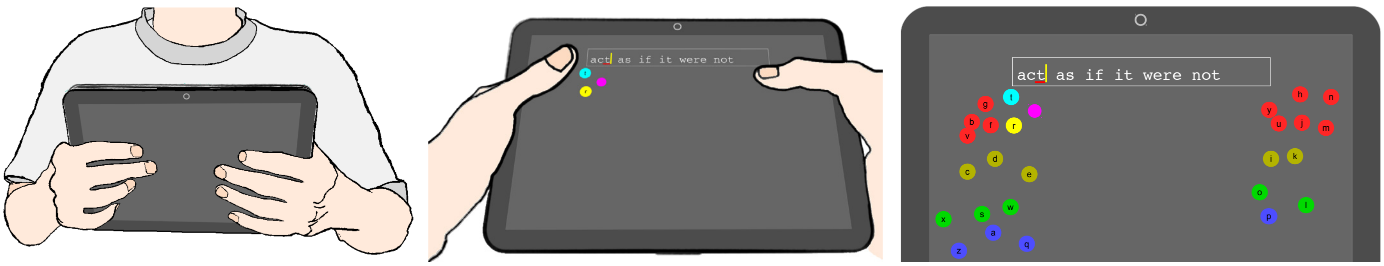

Figure 1. Sandwich Keyboard is gripped from its sides such that the thumbs type on the front side (middle) and the other fingers on the back side (left). Key targets of the keyboard in the back can be projected to the front to aid in learning the technique (right). Colors denote finger assignment.

ABSTRACT
This Note introduces a keyboard design that affords ten-finger touch typing by utilizing a touch sensor on the back side of a device. Previous work has used physical buttons. Using a touch sensor has the benefit that it retains the form factor and does not insist on a peripheral device. Moreover, any layout can be used. However, it is difficult to hit targets on a flat surface with no haptic feedback. Sandwich Keyboard is a prototype that folds any three-row keyboard layout and thus, by retaining the finger-to-letter assignment, supports transfer. Sandwich Keyboard includes an algorithm for constant adaptation of key targets in the back. We also learned that the detection of key presses from finger release enhances the performance of touch-typing on a multitouch sensor. After eight hours of training, experienced typists of the QWERTY and of the Dvorak Standard Keyboard (DSK) layout reached 26.1 and 46.2 wpm, respectively. We discuss improvements necessary for further increasing both speed and accuracy.

Author Keywords
Text entry; touch-typing; touch sensor input; mobile devices; back-of-device interaction

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Input devices and strategies

General Terms
Human Factors; Design; Measurement.

INTRODUCTION
A well-known bottleneck of mobile devices using touchscreens is the low typing speed, leading HCI researchers to examine the use of multiple fingers in typing. Use of the two thumbs has not yielded the desired improvements; the highest reported rate is 37.1 words per minute (wpm) [5]. The question arises of whether other fingers can be employed. A recent study showed 30.7 wpm on a with a two-handed typing technique that requires a stable surface and special posture, thus restricting the user's mobility [4]. On a smaller form factor, experts reached 23.2 wpm in a six-finger technique [9].

This Note contributes to an emerging line of work investigating the potential of *back-of-device* input [1, 8, 11, 12, 13]. The advantage for text entry is that, on a device large enough, all ten fingers can be employed, which potentially improves performance. The device can be held in both hands while the user is moving, and more screen space is saved for the foreground application. Earlier designs featured *physical buttons* attached to the back [3, 7]. Their potential advantage is that they provide tactile feedback for key presses and allow *preparatory finger movements*: a finger can rest on its next key target, waiting for its turn [6]. However, first user tests showed discouragingly low rates of 15 wpm for *RearType* [7] and, on the smartphone form factor, *Back Keyboard* [3]. An important breakthrough was *folding* the layout of the familiar three-row button keyboard such that the finger-to-letter assignment remains the same [7]. This could facilitate positive transfer from a user's existing skills with a physical keyboard. In particular, touch typists, using all ten fingers, could benefit.

We extend the study of folded keyboards on mobile devices to the case wherein an additional *touch sensor*, in the back, tracks the fingers (see Figure 1). *Sandwich Keyboard* consists of a double-touchscreen tablet. It is grasped in the manner of a sandwich, such that the alphabetic characters are typed with four fingers per hand while the thumbs are responsible

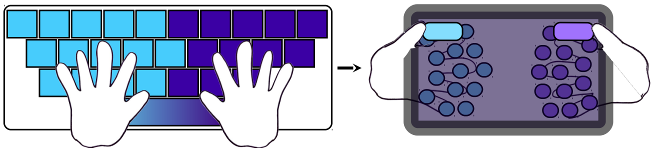

Figure 2. ***Keyboard folding*** **refers to translating and rotating the keyboard halves to the back side of a tablet such that the finger-to-letter assignment is retained. Unlike physical keyboards, the touch sensor enables adapting key targets to individual-specific hand dimensions and grips.**

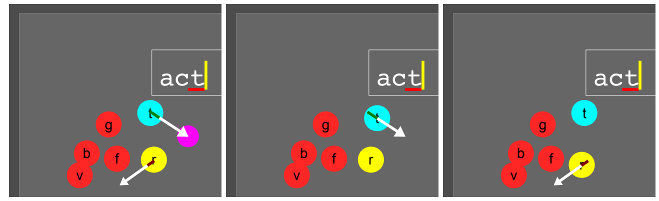

Figure 3. Key targets float from their original position (left) toward (middle) or away from (right) a tap position (magenta) in order to adapt to the user's posture. The intended key was *t*; the one hit was *r*.

for the spacebar and backspace. The benefit of a touch sensor is that it allows any grip, maintains the familiar and practical form factor of mobile tablets. Unlike physical keys in the back, touch sensors are compatible with other touch-based back side interaction schemes, e.g. [11]. However, this system lacks haptic feedback, which is thought to be important for typing performance also in the back-of-device case [7].

This paper describes a design incorporating three techniques which we learned improve typing performance:

- *Adaptive keyboard folding*: We fold the keyboard to retain the familiar finger-to-letter assignment [7]. To make this usable for a touchscreen device, we employ soft keyboard adaptation [2] such that the key targets adapt to the user's hand dimensions and preferred grip *on-line* (i.e., during typing). This improves robustness for anatomical and contextual variations in grip location and hand shape.
- *Projection of tap locations on a virtual keyboard*: To assist in learning of the key locations at the back, we provide a projection of the finger-tap points on the back side. After a user has learned the key locations, the projection can be removed, freeing more space for the foreground application.
- *Key-press event*: We register key events upon finger *release* rather than *touch-down*. This common mode on soft keyboards is even more advantageous in our touch-typing system, because it allows for a touch sensor version of preparatory movements on the flat surface: a finger waiting for its turn can move to and already rest on its next target [6] without causing an erroneous tap.

Before this study, it was not known whether touch sensors, lacking haptic feedback, are suitable for ten-finger typing. The six typists in our user study, after eight training sessions, averaged on 36.1 wpm, which was 64% of their rates with physical keyboards. We conclude with a discussion of how to improve the technique.

DESIGN AND IMPLEMENTATION

Sandwich Keyboard was built by attaching two tablets to each other. The back device can track the touches of the digit fingers and communicate them to the front-side tablet, which constructs the text and can show projections of touches and virtual keys (see Figure 1).

Using a touch sensor in the back poses two design challenges: 1) finding an effective way for the user to move the fingers in the back such that efficient typing is enabled and 2) developing a mechanism whereby the device infers the intended letter reliably. To pursue high entry rates, the user must be able to hit the key targets in the back reliably without being guided by visual or tactile feedback.

Keyboard Folding

Folding a keyboard refers to rotation and translation of the keyboard halves in a manner that allows the typist to retain the familiar finger-to-letter mapping of a physical keyboard, as shown in Figures 1 (right), and 2. We studied possible grips to find one that is comfortable, is stable, and allows a wide enough range of movement for all fingers. We assume that the typist can clasp the device anywhere on its *lateral edges*, thus fixing it between the hands while sustaining a high level of finger mobility and avoiding overlapping of fingers (see Figure 1, left). This grip allows for high velocities in finger movement, but recognition of tap locations as letters is still problematic, since the position of the fingers can vary both laterally and in the extension / flexion direction.

Key Detection and Adaptation

In our solution, whenever a finger is released from the back side sensor (rather than on a touchdown), Sandwich Keyboard uses the closest key target, calculated via Euclidean distance between the tap point and the center of the target, to determine which letter is being entered (Figure 3, left). The system moves key targets to *adapt* the keyboard to gradual changes in posture and typing behavior. An intentionally hit target moves toward the tap location (Figure 3, center) while an erroneously hit target moves away from the touch's location (Figure 3, right) to resolve overly close clusters of key targets. Movement of a key means that if $\overline{k} = (x_k, y_k)$ is the location of the key target's center and $\overline{t} = (x_t, y_t)$ is the tap point's location, then the new key target location will be $\overline{k} + c \cdot (\overline{t} - \overline{k})$. Positive values for c mean a movement toward the touch, whereas negative values effect a movement away from it. Simulations from preliminary training data have shown $1/4$ (toward) and $-1/6$ (away from) to be good first choices for the value of c, since they yielded relatively low error rates. In our user study, the stimulus informs this adaptation about whether the key hit was also the intended one. In real typing scenarios, falsely hit keys are detected, *post hoc*, from subsequent backspace entries.

Projection of the Back Side

To allow a typist to learn the new motor skill more quickly, we project both the virtual keys and the user's tap locations in the back to the display in the front (magenta circles in Figures 1 and 3). Intended keys (demanded by the current stimulus) can also be highlighted. Projection can be turned off once the visual feedback is no longer necessary. In our user study, users were ready to turn projection off after four hours of practice. Every user switched to this "blind" mode easily and soon came to prefer it over the mode with projection.

System Design

We used *Kerafol*[1] *Gluey Soft* adhesive pads to attach two *Toshiba AT-200* tablets running Android 4. These 10-inch tablets were chosen because they have a flat back and weigh only 535 g. This gives the sandwich device a total weight of 1090 g and dimensions of 226×155×7 mm; the touch-sensitive area is 208×137 mm, and the touch-insensitive rim is 18 mm wide. The device is relatively heavy, but its weight is appropriately distributed to allow being held comfortably in the hands. Communication between the tablets was done via *Bluetooth*, with the travel times for event messages established to be constantly 4.5 ms, which enabled typing without noticeable lag.

EVALUATION

To evaluate Sandwich Keyboard, we arranged training consisting of eight one-hour sessions over a period of 14 days. Sessions comprised sections of warm-up exercises, practice drills, and performance tests. We placed breaks between the sections, where the subjects could lay down the device. At the beginning of each section, to cope with inevitable posture changes after taking up the tablet again, we employ an *initialization* phase over 5–10 sentences. In this phase, every key press is regarded as intentional, causing the key hit to move toward the touch, except when corrected afterward via a backspace. During drills and tests, standard adaptation was employed, and backspace disabled. From the third session onward, the highlighting of the intended key was switched off, and from the fourth session on, the projection of the virtual keyboard was switched off, so only the tap points remained projected, as magenta circles. Errors were always reported back to the typist. See Table 1 for an overview of the sessions. In the first two sessions, key locations were practiced. The key to hit for the current character in the stimulus phrase was highlighted (in cyan; see Figures 1 and 3). *Testing* (i.e., evaluation of a subject's current typing performance) was done in every session but the first.

Participants: As the touch sensor allows folding of any three-row layout, we recruited three touch-typists of DSK and QWERTY, each. We included DSK in order to better understand the effect of layout. The two groups are not directly comparable. The DSK typists' mean age was 25, and the QWERTY typists' mean age 27 years. The DSK typists had, on average, seven years of experience in touch-typing with physical keyboards, while QWERTY typists had 11. One QWERTY typist was left-handed; all other participants were right-handed. The participants were compensated at € 10,– per hour.

Task and Materials: Subjects clasped the device at a predetermined position at its lateral edges, tilted downwards about 60° and rested on a foam cushion, to alleviate the relatively heavy weight and decrease fatigue. We used the task of *transcription*, both in training of the subjects and when testing performance: one phrase at a time was presented on the front display (see Figure 1, middle), and the user had to type it quickly and accurately. The software encouraged speeding up when error rates were low (below 5%) and slowing down with high error rates (above 6%). As the corpus, we used randomly sampled phrases from the *MobileEmail* corpus [10]. The categories of drills comprised common bigrams, words, and sentences in the English language. To test performance, we used a subset of phrases not used in the training set. Toward the end of the fifth session, we explained to the participants the strategy of preparatory finger movements (see above) and instructed them to deploy that strategy consciously. Five of six participants found that very difficult at first, but all six participants mastered it without difficulty in the sixth session.

Measurement: We computed wpm as $\frac{|T|-1}{S} \cdot \frac{60}{5}$, where $|T|$ is the number of characters in both the stimulus and the transcribed string, and S is the time needed to transcribe the phrase, in seconds (cf. [2]). As the error rate, we used the *Damerau-Levenshtein* edit distance.

Results

The average typing performance with a *physical* (notebook) keyboard was 54.7 wpm at 3.6% errors for DSK typists and 58.7 wpm at 2.7% errors for the QWERTY typists, all using the corresponding layout. This constitutes the baseline level.

The development of the typists' performance with Sandwich Keyboard is presented in Figure 4. The DSK typists reached a level of 46.2 wpm at 9.8% errors in the eighth session, while the QWERTY typists reached 26.1 wpm at a 12.6% error rate. A caveat in comparison of the two keyboards is that one QWERTY typist had to cease participation after six sessions. However, the rates of the DSK typists were already 60% higher at that point, suggesting superiority of DSK over QWERTY in the current technique.

The preparatory movements we encouraged led to a noticeable boost in typing speed in the sixth session as well as to higher satisfaction among the subjects.

The six users were generally positive about the folded-keyboard design's features. After six hours, five participants found typing on the sandwich device to feel easy and like touch-typing. Two DSK typists in our study stated that the most overloaded fingers (the index fingers) are the most difficult to use in this application. Two subjects were unsatisfied with operating the spacebar by means of the thumb at the front while everything else is at the back side. One DSK typist suffered from delicate finger joints and said that on the flat screen it is less painful to force herself to speed up than on a physical keyboard.

Day	Important session features	Goal
1	Explanations, drill demos	Understand the technique
2	Drills, first test	Familiarization
3	Key-highlight hidden	Try to type fluently
4–5	Drills, projection hidden	Acquire accuracy
6–7	Drills, projection hidden	Acquire speed
8	Drills, final test	Measure performance

Table 1. Sessions in the study

[1] `http://www.kerafol.com/`

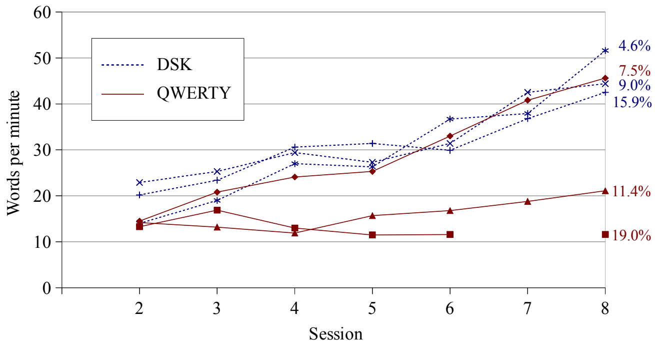

Figure 4. Typing performance with Sandwich Keyboard as a function of session. Error rates from the final session are given as percentages.

DISCUSSION

Sandwich Keyboard was designed to facilitate the transfer of a touch typist's existing skill to a mobile device. Our data suggest that a typist can adopt the new technique and approach a satisfactory level of performance within 6–8 hours. After eight hours of training (explanation, drills, and tests), the wpm figures reached 36.1, which is 64% of the subjects' performance with a physical keyboard. The trend indicates that the typists were still developing in the eighth session. We conclude that learning folded keyboards is fast, since they retain the finger-to-key assignment found with physical keyboards; only the wrist posture and finger movement range change. To support this idea explicitly in design, we presented three techniques that build on previous work: 1) adaptive folding, 2) key-press events detected from finger release, and 3) projection of back side taps on the front side. Our user study suggests that haptic feedback is not absolutely necessary for decent typing performance, even when one employs a naïve recognition algorithm such as ours.

The DSK typists reached surprisingly better performance: their average was 46.2 wpm, which was 84% of their performance with a physical keyboard. One reason could be that QWERTY generally requires more finger-travel movement and, especially, much greater change in finger extension / flexion, because DSK involves more typing on the home row. Anecdotally, the first author of the paper, after 12 days of testing, could reach 70.5 wpm with an error rate of 2.9% with the DSK layout, beating his baseline of 64 wpm (2.9% errors) with the corresponding physical keyboard.

Despite the good performance reached, the results leave room for further improvements still. Recent strategies to improve recognition robustness and thus typing accuracy on flat panels employ machine-learning algorithms [2]. Many of the remaining errors were caused by suboptimal recognition rather than actual movement errors by the typist. While our adaptive recognizer copes well with gradual posture changes, it does not respond gracefully to abrupt change, such as those after putting the tablet down and picking it up again. We believe that interaction and recognition can be adjusted to suit these situations. Dynamic (programmable) tactile feedback mechanisms (e.g. [14]) could improve finding the tap locations at the back while typing. We believe that with such changes and more practice, a rate of 50 wpm is achievable.

ACKNOWLEDGMENTS

This work was supported by the Academy of Finland, the EIT ICT Labs, and the Cluster of Excellence for Multimodal Computing and Interaction at Saarland University.

REFERENCES

1. Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, ACM (2009), 1923–1932.
2. Findlater, L., and Wobbrock, J. Personalized input: Improving ten-finger touchscreen typing through automatic adaptation. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, 2012), 815–824.
3. Kim, H., Row, K., and 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, ACM (New York, 2012), 15831588.
4. Li, F. C. Y., Guy, R. T., Yatani, K., and Truong, K. N. The 1line keyboard: A QWERTY layout in a single line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, ACM (New York, 2011), 461470.
5. Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, ACM (New York, NY, USA, 2013), 27652774.
6. Salthouse, T. A. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin; Psychological Bulletin 99*, 3 (1986), 303.
7. Scott, J., Izadi, S., Rezai, L. S., Ruszkowski, D., Bi, X., and Balakrishnan, R. RearType: Text entry using keys on the back of a device. In *Proceedings of the 12th Conference on Human–Computer Interaction with Mobile Devices and Services*, MobileHCI '10, ACM (New York, 2010), 171180.
8. Shen, E.-L., Tsai, S.-S., Chu, H.-H., Hsu, J. Y.-j., and Chen, C.-W. Double-side multi-touch input for mobile devices. In *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, ACM (2009), 4339–4344.
9. Southern, C., Clawson, J., Frey, B., Abowd, G., and Romero, M. An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, MobileHCI '12, ACM (New York, NY, USA, 2012), 317326.
10. Vertanen, K., and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. In *MobileHCI '11: Proceedings of the 13th International Conference on Human–Computer Interaction with Mobile Devices and Services*, ACM (2011), 295–298.
11. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, ACM (New York, 2007), 269–278.
12. Wobbrock, J. O., Myers, B. A., and Aung, H. H. The performance of hand postures in front- and back-of-device interaction for mobile computing. *International Journal of Human–Computer Studies 66*, 12 (2008), 857–875.
13. Wolf, K., Müller-Tomfelde, C., Cheng, K., and Wechsung, I. Does proprioception guide back-of-device pointing as well as vision? In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, ACM (New York, NY, USA, 2012), 17391744.
14. Xu, C., Israr, A., Poupyrev, I., Bau, O., and Harrison, C. Tactile display for the visually impaired using TeslaTouch. In *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems – CHI EA '11* (Vancouver, BC, Canada, 2011), 317.