A Japanese Software Keyboard for Tablets That Reduces User Fatigue

Teppei Yajima Hiroshi Hosobe
Faculty of Computer and Information Sciences
Hosei University
Tokyo, Japan
Email: hosobe@acm.org

***Abstract*—Increasingly more people are using tablets for various tasks such as browsing websites, reading and writing email messages, making notes, and even creating business documents, for which they often need to enter text. When people enter text with tablets, they usually use software keyboards on the touch screens of the tablets. Unlike hardware keyboards that normally allow users to put their fingertips on keys, software keyboards require users to keep their fingertips in the air except when they actually perform input by tapping the screens, which may cause fatigue to the users. In this paper, we propose a Japanese software keyboard for tablets that reduces user fatigue. Our software keyboard allows its user to put his/her fingertips on the touch screen by distinguishing the user's intended tapping from his/her touching the screen without intending to input. We evaluated the keyboard by conducting a user study that compared it with a software keyboard based on normal tapping. The results of the user study show that our software keyboard caused less fatigue to the users in several regions of their bodies than the normal keyboard, although it resulted in a longer average time length and a worse average error rate of the users' text entry.**

***Keywords*-software keyboard, text entry, touch screen, user fatigue**

I. Introduction

Increasingly more people are using tablets for various tasks such as browsing websites, reading and writing email messages, making notes, and even creating business documents, for which they often need to enter text. If they used normal personal computers (including laptops) for such tasks, they could use hardware keyboards to enter text. However, most tablets themselves are not equipped with hardware keyboards. Although such tablets normally can be connected with external hardware keyboards, hardware keyboards may degrade their mobility, and therefore many people use tablets without hardware keyboards. Therefore, when people enter text with tablets, they usually use software keyboards on the touch screens of the tablets.

Software keyboards require users to keep their fingertips in the air except when they actually perform input by tapping the screens. In this regard, they are a contrast to hardware keyboards that normally allow users to put their fingertips on keys. This property of software keyboards may cause additional fatigue to the users. For example, Kim et al. [9] reported that, compared to hardware keyboards, a software keyboard decreased the comfort of users and increased the activity of their shoulder muscles.

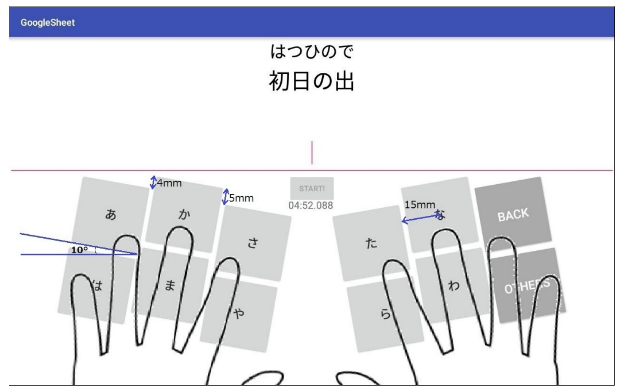

Figure 1. Proposed Japanese software keyboard.

In this paper, we propose a new Japanese software keyboard for tablets that reduces user fatigue (Figure 1). Our main idea is that the software keyboard should allow its user to put his/her fingertips on the touch screen of the tablet, without requiring the user to keep the fingertips in the air. For this purpose, our software keyboard distinguishes the user's intended tapping from his/her touching the screen without intending to input. Instead of using a pressure-sensitive touch screen, we use a 10.1-inch capacitive-sensitive 10-point multi-touch screen that is widely used for tablets. Because of this, we use the durations of the user's touches to distinguish his/her intended tapping from others.

Our software keyboard consists of 12 keys and is specialized in the entry of Japanese kana characters in the same way as the software keyboard previously developed by Takei and Hosobe [20]. It should be emphasized that our keyboard is focused on the reduction of user fatigue, rather than on the entry of Japanese kana characters, on which Takei and Hosobe's keyboard was focused especially to enable touch typing. The reason why we propose a Japanese software keyboard instead of an English one is that Takei and Hosobe's keyboard can be constructed with 12 keys, which satisfies the conditions that we found from the preliminary experiments that we report in Section IV. In other words, the basic idea behind our software keyboard that we introduce

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.

in this paper will be applicable to an English keyboard if we can deal with this small number of keys by using another approach.

We evaluated our software keyboard by conducting a user study that compared it with a software keyboard based on normal tapping. The results of the user study show that our software keyboard caused less fatigue to the users in several regions of their bodies including their shoulders than the normal keyboard. However, the results also indicate a trade-off between user fatigue and performance: our software keyboard resulted in a longer average time length and a worse average error rate of the users' text entry than the normal keyboard.

The rest of this paper is organized as follows. After presenting related work in Section II, we briefly describe Japanese kana characters and Takei and Hosobe's software keyboard in Section III. In Section IV, we report the preliminary experiments that we conducted to determine the basic design of our software keyboard. Then, in Section V, we propose our software keyboard. In Section VI, we report the experiment that we performed to evaluate it, and discuss it in Section VII. Finally, we describe conclusions and future work in Section VIII.

II. Related Work

There has been research on potential problems of software keyboards to users. Kim et al. [9] compared a software keyboard, a laptop computer's hardware keyboard, and a desktop computer's hardware keyboard in regard to typing forces, muscle activity, and typing performance of users. Their experiments showed that, although a software keyboard needed lower typing forces and lower finger muscle activity, it resulted in decreased typing performance, decreased self-reported comfort, and increased shoulder muscle activity. Kim et al. [10] also investigated how key sizes of software keyboards affected typing force exposures, muscle activity, wrist posture, comfort, and typing productivity. They showed that software keyboards with a key size less than 16 mm might be too small due to slower typing speed, higher static shoulder muscle activity, greater wrist extension, and lowest subjective preferences. Lin et al. [14] studied how three usage positions (desk, lap, and bed) during typing with three tablet software keyboards (standard, wide, and split) affected the upper-body kinematics and discomfort of users and the usability of the keyboards. They showed that such use of tablets might expose users to greater risks of musculoskeletal symptoms. Unlike these studies that examined problems with existing software keyboards, our work is focused on exploring a new software keyboard that especially aims at reducing user fatigue.

There has been research on user performance of software keyboards. Sears et al. [18] investigated how key sizes of software keyboards affected typing speed and error rates. Hasegawa et al. [7] examined how the ages of users and the use of only dominant, only non-dominant, and both hands affected their performance in typing with software and hardware keyboards used for tablets. Yanai and Karashima [22] studied user performance of software keyboards especially from the viewpoint of two-dimensional distributions of positions of key touches. These studies differ from our work especially in that they were focused on investigating the user performance of existing software keyboards.

There has been research on new software keyboards for tablets and other devices. Himberg et al. [8] proposed personalizing the layout of a 3-by-3 software keyboard by on-line learning. Findlater et al. [2] proposed personalizing the layout of a QWERTY software keyboard and integrating multi-touch gestures with the keyboard. Go and Endo [4] developed a software keyboard that enabled the user to perform precise input even when their fingertips touched on the boundary of keys. Kuno et al. [11], [12] developed a software keyboard whose layout could be changed to fit users' hands. Hakoda et al. [6] proposed a portrait-style QWERTY software keyboard for touch screen devices. Shibata et al. [19] proposed a software keyboard called DriftBoard for ultra-small touch screens like smart watches. Lenovo [13] developed a keyboard called halo that allowed its user to type on a specific touch panel without hardware keys (that was, strictly speaking, not a software keyboard). Although these studies also proposed new software keyboards, they were different from our work that proposes a software keyboard especially aiming at reducing user fatigue.

There has been research on multi-stroke gesture-based software keyboards for tablets and other devices, on which the users drag their fingertips to enter words. Unlike ordinary tapping-based software keyboards, such multi-stroke gesture-based keyboards typically do not require the users to lift their fingertips while entering individual words. Rick [15] examined the influence of different keyboard layouts for a multi-stroke gesture-based keyboard for an interactive tabletop. Bi et al. [1] proposed two interaction techniques for a bimanual multi-stroke gesture-based keyboard for a tablet. Although such multi-stroke gesture-based keyboard might be effective for reducing user fatigue, the focus of these studies was different from that of our work; they were focused on the improvement of the performance of the users' gesture input.

Sax et al. [17] developed an ergonomic software keyboard by assigning home keys to the fingers of a user and forming groups for the home keys, which enabled the key layout to fit his/her hands. In addition, they proposed allowing the user to put the fingertips on the screen by sensing their pressure against the screen although they did not actually implement this feature. In some sense, their work is similar to our work, but their approach is not applicable to ordinary capacitive-sensitive touch screens that do not sense pressure.

There has been research on software keyboards for the Japanese language. Fukatsu et al. [3] and Hakoda et al. [5]

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.

proposed software keyboards for smart phones that enabled eyes-free Japanese kana input. Sakurai and Masui [16] proposed a QWERTY software keyboard for tablets that enabled Japanese kana input by using flick operations. Takei and Hosobe [20] developed a 2-by-6-key software keyboard for tablets that enabled the user to do the touch typing of Japanese kana characters usually with two key strokes. It should be noted that, although our work also treats Japanese kana characters, it is not focused on the development of a new Japanese software keyboard as already mentioned in Section I.

III. Preliminaries

In this section, we briefly describe Japanese kana characters and Takei and Hosobe's 2-by-6-key Japanese software keyboard.

A. Japanese Kana Characters

Japanese text uses two kinds of characters, i.e., Chinese characters and *kana* characters. While a Chinese character typically has a meaning, a kana character does not; instead, a kana character is associated with a speech sound. There are two kinds of kana characters called hiragana and katakana. Although they are used for different purposes, they correspond to each other.

There are approximately 50 basic kana characters, which are further divided into 10 groups that are ordered, each of which typically consists of five characters. The first group is special because its five characters indicate five vowels that are pronounced "a," "i," "u," "e," and "o." The other nine groups are associated with the basic consonants, "k," "s," "t," "n," "h," "m," "y," "r," and "w." A kana character in these nine groups forms the sound that combines a consonant and a vowel. For example, the five characters of the "k" group are pronounced "ka," "ki," "ku," "ke," and "ko." Certain groups have variants called dakuon and handakuon, and certain characters have variants that are written in smaller shapes.

B. Takei and Hosobe's Software Keyboard

Takei and Hosobe [20] proposed a 2-by-6-key software keyboard for entering Japanese kana characters on a tablet's touch screen. They basically use two key strokes for one basic kana character. The first stroke is used to select one from the 10 groups of basic kana characters, and the second stroke is used to select the basic kana character. As shown in Figure 2, the 2-by-5 white keys on the left are assigned to this task, and the 2-by-1 gray keys on the right are assigned to the special keys called "back" and "other."

Their keyboard supports 85 graphic characters (i.e., 81 kana characters, three Japanese special characters, and "space") and two control characters ("backspace" and "enter"). The 80 characters called *2-stroke characters* can be entered with two strokes. The six characters called *3-stroke characters* can be entered with three strokes. The other

Figure 2. Takei and Hosobe's software keyboard [20].

Table I
Characters (except "backspace") supported by Takei and Hosobe's software keyboard [20]

Group	Upper	Lower 1	Lower 2
あ	あいうえお	ぁぃぅぇぉ	
か	かきくけこ	がぎぐげご	
さ	さしすせそ	ざじずぜぞ	
た	たちつてと	だぢづでど	っ
な	なにぬねの		
は	はひふへほ	ばびぶべぼ	ぱぴぷぺぽ
ま	まみむめも		
や	や ゆ よ	ゃ ゅ ょ	
ら	らりるれろ		
わ	わ を ん	ゎ	
other	enter 、。ー space		

character, "backspace," can be entered with one stroke using the "back" key.

To enter a character X, the user performs the following operations. (1) If X is "backspace," the user touches the "back" key. (2) Otherwise, the user touches the key corresponding to the group of X, and next does the following. (2a) If X is a 2-stroke character, the user touches the key corresponding to X. (2b) Otherwise (i.e., X is a 3-stroke character), the user touches the "other" key and then the key corresponding to X.

In Table I, 2-stroke characters are shown in the "Upper" and "Lower 1" columns, and 3-stroke characters are shown in the "Lower 2" columns. Characters in the "Upper" columns are shown on the upper white keys at step (2) including substeps (2a) and (2b), and characters in the "Lower 1" and "Lower 2" columns are shown on the lower white keys at substeps (2a) and (2b) respectively.

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.

Figure 3. Preliminary experiment 1 for measuring possible positions of fingertips on a touch screen.

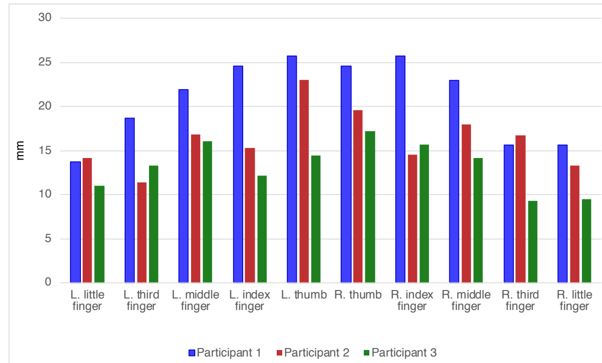

Figure 4. Results of preliminary experiment 1 on possible moving distances of fingertips on a touch screen.

IV. Preliminary Experiments

In this section, we report two preliminary experiments that we conducted to determine the basic design of our software keyboard.

A. *Preliminary Experiment 1*

In the first preliminary experiment, we examined how much users can move their fingertips on a touch screen. This aimed at determining an appropriate key layout for allowing users to manipulate it without difficulty in moving their fingers. It should be noted that, since we use (instead of a pressure-sensitive touch screen) a capacitive-sensitive touch screen that is popular among current tablets, users cannot manipulate the touch screen with their fingernails.

We recruited three participants ranging from 21 to 22 in age. Each participant sat on a chair and started the experiment with their 10 fingertips put on the touch screen (Figure 3). We asked the participant to move their fingertips freely on the touch screen as far as he/she found no difficulty in it. We measured the positions of the fingertips while he/she was moving the fingertips; we did not measure their positions while there were less than 10 fingertips touching the screen. For this experiment, we implemented an Android application in Java, and used an ASUS ZenPad 10 Z300M tablet equipped with a 10.1-inch 10-point multi-touch screen.

Figure 4 shows how much the participants were able to move their fingertips in this experiment. We can see that the thumbs were able to move the most and the little fingers were able to move the least. The average distances of how much the fingertips moved were approximately 20 mm, 18 mm, 18 mm, 14 mm, and 13 mm for the thumbs, the index, the middle, the third, and the little fingers respectively. Also, we observed that the fingertips of the participants had disturbed each other due to the size constraint of the 10.1-inch touch screen. It means that even a user who has large hands cannot move their fingertips much on the touch screen of this size.

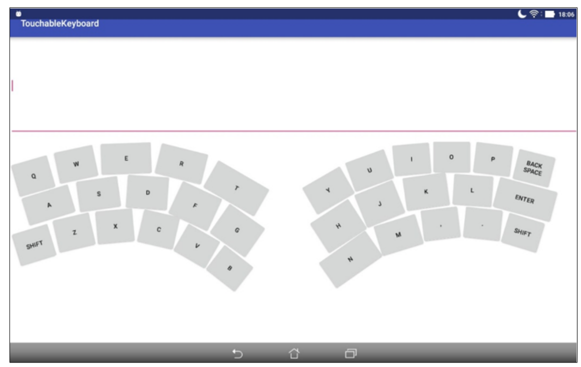

Figure 5. Prototype QWERTY software keyboard.

B. *Preliminary Experiment 2*

We conducted a second preliminary experiment on the accuracy of the multi-touch sensing of the tablet. Before this, we implemented a prototype QWERTY software keyboard based on our basic idea of allowing users to put their fingertips on the touch screen (i.e., the QWERTY version of the software keyboard that is proposed in the next section) as shown in Figure 5. However, we found that the keyboard frequently had recognized erroneous input. More specifically, such errors occurred at locations close to the fingertips that the user moved intentionally for input. We thought that the errors were caused by the insufficient accuracy of the multi-touch sensing of the tablet, and therefore we conducted this experiment.

The experiment was done by the first author. We did not recruit third-party participants because the purpose of this experiment was not to examine user performance but was to examine the performance of the multi-touch sensing of the tablet. During the experiment, he put the fingertips of both of his hands on the touch screen and moved them freely. When the number of the recognized fingertips became less than 10, we measured the distance between the fingertip that disappeared and the one closest to it.

The result of this experiment showed that, when the

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.

distance between two fingertips became less than approximately 3 mm, the touch screen recognized them as one fingertip. This result indicates that we need sufficiently large keys to realize a software keyboard based on our idea. The QWERTY layout is not appropriate for this purpose because its keys cannot be sufficiently large on an approximately 10-inch touch screen.

V. Proposed Keyboard

We propose a Japanese software keyboard for tablets that reduces user fatigue. Our main idea is that the software keyboard should allow its user to put his/her fingertips on the touch screen of the tablet, without keeping the fingertips in the air, which is required by ordinary software keyboards for tablets. For this purpose, our software keyboard needs to recognize whether the user actually intends to input when he/she touches the screen. We realize this by using how long the user touches the screen: if the user touches the screen for less than 140 milliseconds, the keyboard regards the user as tapping, i.e., doing intended input; otherwise, the keyboard does not treat it as tapping. Here we use the threshold of 140 milliseconds by following Kuno et al.'s work [12], where they used this threshold to distinguish tapping and calibration operations on a touch screen.

Our keyboard consists of 12 keys as shown in Figure 1. These keys are the same as the ones that Takei and Hosobe [20] introduced in their software keyboard (Figure 2). However, the entire layout of the keys in our keyboard is more "ergonomic" than that of Takei and Hosobe. This is necessary because our keyboard allows its user to put their fingertips on the touch screen of the tablet, which makes the fingertips less movable than when the user keeps their fingertips in the air. To determine this key layout, we used the results of preliminary experiment 1 that we described in Subsection IV-A.

Our software keyboard allows its user to enter Japanese kana characters in the same way as Takei and Hosobe's keyboard; that is, the user basically can enter one kana character with two key strokes, firstly by tapping the key corresponding to the character's group, and then by tapping the key corresponding to its vowel.

VI. Experiment

In this section, we report the experiment that we conducted to evaluate the proposed keyboard.

A. Procedure

We conducted an experiment to evaluate the proposed keyboard. We compared the following two conditions: one where users used the keyboard by keeping their fingertips in the air (called the *normal* keyboard below for brevity); the other where users used it by putting their fingertips on the touch screen (called the *proposed* keyboard below). For this experiment, we implemented both keyboards in Java as Android applications for an ASUS ZenPad 10 Z300M tablet equipped with a 10.1-inch capacitive-sensitive touch screen supporting 10-point multi-touch sensing.

We recruited 10 participants ranging from 18 to 24 in age. During the experiment, the tablet was placed on a desk, and the participants sat on a chair. The participants were given words, each of which consisted of five kana characters and was displayed on the upper half of the screen. We asked them to enter the given characters and then to touch the "enter" key to move to a next word. Before the experiment, the participants entered five words for practice. In the experiment, they entered two sets of words, each of which consisted of 60 words. The two sets corresponded to the normal and the proposed keyboard. The participants were asked to keep their wrists in the air when using the normal keyboard; also, they were asked to put their wrists on the desk when using the proposed keyboard.

Each participant used both keyboards. To avoid an order effect, we divided the participants into two groups, and assigned the two keyboards to the two groups in the different order. At least one-hour interval was given to each participant between the experiments for the two keyboards.

We measured average time lengths for characters and average error rates for words. Average error rates were computed in the same way as [5], which is as follows: compare the character sequence entered by a participant with the word presented on the screen; count the total of wrong characters, unnecessary characters, and missing characters; compute the average error rate as the average division of this total by the number of the characters in the presented word.

The participants answered to a questionnaire after entering each set of words. We used a questionnaire for assessing regions of fatigue in the body [21]. More specifically, immediately after the experiment, we assessed the degrees of fatigue in the neck, the shoulders, the upper arms, the elbows and forearms, the wrists and hands, the upper back, and the lower back by using the four scales 0 (none), 1 (weak), 2 (neither weak nor strong), and 3 (strong).

B. Results

Table II shows the results of the measurements that we obtained from the experiment. The proposed keyboard needed a longer average time length than the normal keyboard. We think that there were two main causes for this result. One main cause was that the participants needed more movements. When using the normal keyboard, the participants needed two movements for one tapping operation; that is, moving down the necessary fingertip on the touch screen and then moving it up. By contrast, when using the proposed keyboard, the participants needed three movements for one tapping operation and one movement after it; that is, moving up the necessary fingertip from the touch screen, moving it down and up for actual tapping, and finally moving down

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.

Table II
AVERAGE TIME LENGTHS FOR CHARACTERS, AVERAGE ERROR RATES FOR WORDS, AND AVERAGE NUMBERS OF TIMES "BACKSPACE" WAS USED IN THE CASES OF THE NORMAL AND THE PROPOSED KEYBOARD

Keyboard	Time (sec)	Error (%)	Backspace
Normal	1.9	0.33	29.1
Proposed	2.3	0.50	32.5

it again on the screen. The other main cause was that the fingertips on the touch screen hid the keys, due to which the participants needed more time for checking the keys. Also, the proposed keyboard resulted in a worse average error rate and a larger average number of times the "backspace" was used. We think that the proposed keyboard caused wrong recognition of tapping due to the movements of fingertips put on the touch screen.

Next, we analyzed the trend of errors by examining which key was erroneously tapped by the participants when they needed to tap a certain key. Figure 6 shows the results of this analysis, where the errors that occurred three or more times are shown. Figures 6(a) and 6(b) show the results of the normal keyboard, and Figures 6(c) and 6(d) show those of the proposed keyboard. In both cases, the first one, i.e., Figures 6(a) and 6(c), show the results of the first strokes, and the second ones, i.e., Figures 6(b) and 6(d), show those of the second strokes. In each figure, "r/w," for example, indicates an error where key "w" (for the "w" group) was erroneously tapped when key "r" (for the "r" group) should have been tapped.

The results indicate that both the normal and the proposed keyboard encountered many errors where a key that was horizontally or vertically next to a correct one was tapped. Also, there were many errors related to the "others" key. For example, key "a" was erroneously tapped instead of "others" (i.e., "others/a" in Figures 6(a) and 6(c)). We think that this kind of errors typically happened when a participant tried to move to a next word. For this purpose, he/she needed to tap "others" and then "enter." In this case, his/her failure to tap "others" would result in this kind of an error because he/she would next tap "a," which would appear at the same position as "enter." Other common errors were related to little fingers. In the case of the proposed keyboard, little fingers sometimes touched keys for third fingers, which caused errors such as "h/others," "h/a," and "others/a."

We also performed a paired t-test on the results of the questionnaire. Table III summarizes the results. The proposed keyboard gained better results in the left and right shoulders, the left and right upper arms, and the lower back than the normal keyboard. However, it yielded worse results in the left and right wrists and hand. We think that, since we asked them to put the wrists on the desk, this condition forced the participants to do unnatural movements of the wrists.

Table III
RESULTS OF THE QUESTIONNAIRE ABOUT THE REGIONS OF FATIGUE IN THE BODY IN THE CASES OF THE NORMAL AND THE PROPOSED KEYBOARD

Region of fatigue	Normal	Proposed	p-value
Neck	1.1	0.70	0.17
L. shoulder	1.7	**0.40**	0.0019
R. shoulder	1.3	**0.40**	0.010
L. upper arm	1.4	**0.50**	0.010
R. upper arm	1.2	**0.40**	0.022
L. elbow & forearm	1.4	1.3	0.68
R. elbow & forearm	1.1	1.2	0.73
L. wrist & hand	**1.0**	1.4	0.037
R. wrist & hand	**0.70**	1.4	0.0013
Upper back	0.80	0.40	0.17
Lower back	1.2	**0.40**	0.0031

VII. Discussion

The experiment showed that the proposed keyboard was better than the normal keyboard in the fatigue of the left and right shoulders, the left and right upper arms, and the lower back of the participants. We think that this result is particularly important. The two previous studies by Kim et al. [9], [10] showed that software keyboard might cause higher shoulder muscle activity. Our result that the proposed keyboard reduced the fatigue of the shoulders suggests the effectiveness of our approach to the inherent problem of software keyboards.

In the experiment, little fingers caused common errors that they sometimes touched keys for third fingers. We think that it was due to the design of the key layout that did not consider distances between little and third fingers. We think that it will be possible to alleviate this problem by improving the key layout.

To develop our software keyboard, we used a capacitive-sensitive touch screen that is widely used for tablets. Because of this, our software keyboard required users to do more finger movements than normal software keyboards. It is a main cause of the worse user performance of our keyboard, and we need to solve this problem. However, it is not easy to construct an alternative method since a capacitive-sensitive screen senses only the positions that fingertips and other parts of hands touch.

As Sax et al. [17] suggested, a pressure-sensitive touch screen might be more suitable also for our purpose. However, we do not think that even this is a perfect solution. A pressure-sensitive touch screen cannot provide the same tactile feedback as hardware keyboard; for example, it cannot be clearly depressed unlike hardware keys. Therefore, it is necessary to investigate how better a pressure-sensitive touch screen-based software keyboard performs than the current capacitive-sensitive screen-based keyboard and how much it is comparable with a hardware keyboard.

Our software keyboard treated the Japanese language instead of English. As already mentioned, this was due to

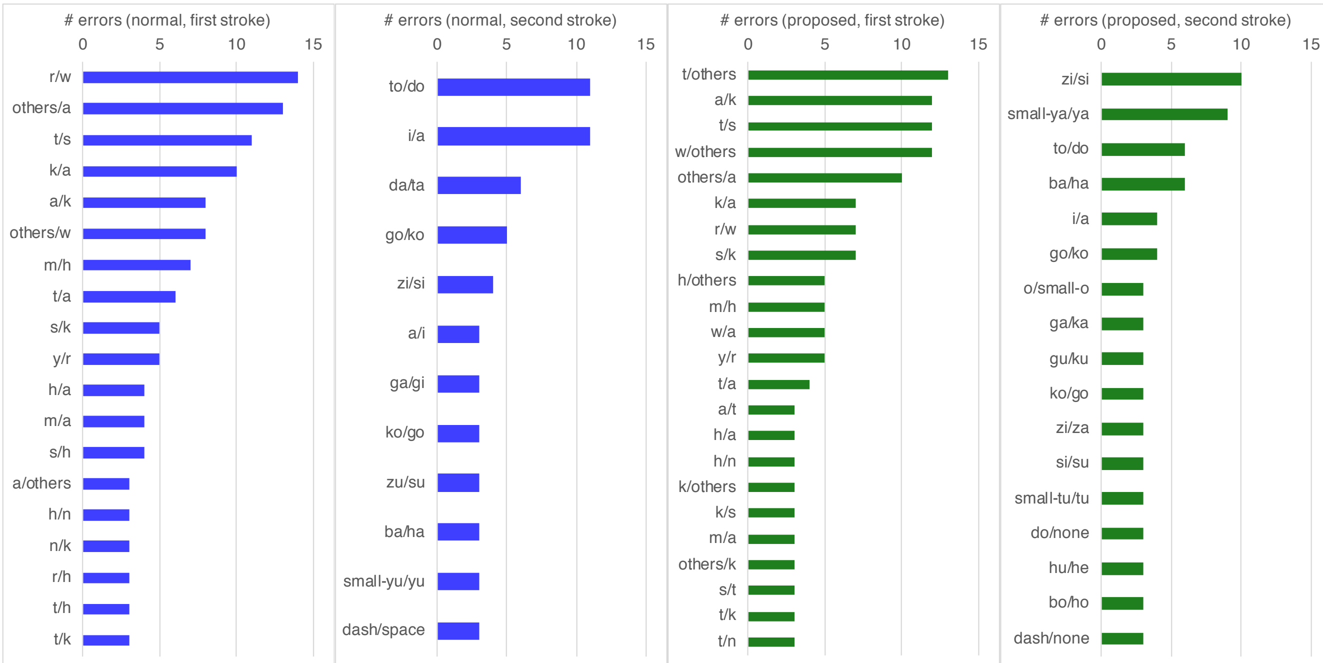

Figure 6. Trend of errors of (a) the first and (b) the second strokes in the case of the normal keyboard and (c) the first and (d) the second strokes in the case of the proposed keyboard.

the results of our preliminary experiments that the touch screen was limited in the possible number and size of keys. We think that it is possible for our approach to treat English by using a special layout other than QWERTY although we need to consider user performance issues.

VIII. Conclusions and Future Work

We proposed a Japanese software keyboard for tablets. With the aim of reducing user fatigue, we designed it in such a way that it allowed the user to put their fingertips on the touch screen as he/she did with an ordinary hardware keyboard. We evaluated the proposed keyboard by comparing the two conditions where the fingertips were in the air and on the touch screen. Although the condition where the fingertips were on the touch screen resulted in a longer average time length and a worse average error rate, it resulted in less user fatigue in the five regions of the body among the eleven that we investigated.

One of our future directions is to resolve the problem of the increased movements of fingertips. Another direction is to explore a key layout that is more suitable for our aim than the current 2-by-6-key layout.

Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Number JP15KK0016.

References

[1] X. Bi, C. Chelba, T. Ouyang, K. Partridge, and S. Zhai. Bimanual gesture keyboard. In *Proc. ACM UIST*, pages 137–146, 2012.

[2] L. Findlater and J. O. Wobbrock. From plastic to pixels: In search of touch-typing touchscreen keyboards. *Interactions*, 19(3):44–49, 2012.

[3] Y. Fukatsu, B. Shizuki, and J. Tanaka. No-look Flick: Single-handed and eyes-free Japanese text input system on touch screens of mobile devices. In *Proc. MobileHCI*, pages 161–170. ACM, 2013.

[4] K. Go and Y. Endo. CATKey: Customizable and adaptable touchscreen. In *Proc. IFIP INTERACT*, volume 4662 of *LNCS*, pages 493–496, 2007.

[5] H. Hakoda, Y. Fukatsu, B. Shizuki, and J. Tanaka. An eyes-free kana input method using two fingers for touch-panel devices. *IPSJ SIG Tech. Rep.*, 2013-HCI-154(6):1–8, 2013. In Japanese.

[6] H. Hakoda, B. Shizuki, and J. Tanaka. Evaluating the performance of a portrait-style QWERTY keyboard for touch-panel devices. *IPSJ SIG Tech. Rep.*, 2014-HCI-156(18):1–7, 2014. In Japanese.

[7] A. Hasegawa, S. Hasegawa, and M. Miyao. Characteristics of the input on software keyboard of tablet devices: Aging effects and differences between the dominant and non-dominant hands for input. *J. Mobile Interact.*, 2(1):23–28, 2012. In Japanese.

[8] J. Himberg, J. Häkkilä, P. Kangas, and J. Mäntyjärvi. On-line personalization of a touch screen based keyboard. In *Proc. IUI*, pages 77–84. ACM, 2003.

[9] J. H. Kim, L. Aulck, M. C. Bartha, C. A. Harper, and P. W. Johnson. Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards. *Applied Ergon.*, 45(6):1406–1413, 2014.

[10] J. H. Kim, L. Aulck, O. Thamsuwan, M. C. Bartha, and P. W. Johnson. The effect of key size of touch screen virtual keyboards on productivity, usability, and typing biomechanics. *Human Factors*, 56(7):1235–1248, 2014.

[11] Y. Kuno, B. Shizuki, and J. Tanaka. Long-term study of a software keyboard that places keys at positions of fingers and their surroundings. In *Proc. HCII*, volume 8008 of *LNCS*, pages 72–81, 2013.

[12] Y. Kuno, B. Shizuki, and J. Tanaka. A software keyboard that places keys at positions of fingers and their surroundings. *IPSJ Journal*, 55(4):1353–1364, 2014. In Japanese.

[13] Lenovo. Lenovo reveals the Yoga Book—the 2-in-1 tablet for productivity and creativity, 2016. Available: http://news.lenovo.com/news-releases/lenovo-reveals-yoga-book-2-in-1-tablet-for-productivity-and-creativity.htm

[14] M.-I. B. Lin, R.-H. Hong, J.-H. Chang, and X.-M. Ke. Usage position and virtual keyboard design affect upper-body kinematics, discomfort, and usability during prolonged tablet typing. *PLoS ONE*, 10(12):e0143585, 2015.

[15] J. Rick. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proc. ACM UIST*, pages 77–86, 2010.

[16] Y. Sakurai and T. Masui. A flick-based Japanese input system for a QWERTY software keyboard. *IPSJ SIG Tech. Rep.*, 2013-HCI-154(5):1–4, 2013. In Japanese.

[17] C. Sax, H. Lau, and E. Lawrence. LiquidKeyboard: An ergonomic, adaptive QWERTY keyboard for touchscreens and surfaces. In *Proc. ICDS*, pages 117–122. IARIA, 2011.

[18] A. Sears, D. Revis, J. Swatski, R. Crittenden, and B. Shneiderman. Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behaviour & Info. Tech.*, 12(1):17–22, 1993.

[19] T. Shibata, D. Afergan, D. Kong, and B. F. Yuksel. DriftBoard: A panning-based text entry technique for ultra-small touchscreens. In *Proc. ACM UIST*, pages 575–582, 2016.

[20] K. Takei and H. Hosobe. A 2-by-6-button Japanese software keyboard for tablets. In *Proc. VISIGRAPP*, volume 2, pages 147–154. SCITEPRESS, 2018.

[21] Working Group for Occupational Fatigue. Assessment of regions of fatigue, 2002. In Japanese. Available: http://square.umin.ac.jp/of/service.html

[22] Y. Yanai and M. Karashima. Research on two dimensional touched position distributions of the touch screen QWERTY keyboard. In *HCII Posters*, volume 373 of *CCIS*, pages 561–565, 2013.

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/COMPSAC.2018.00054.