

A Japanese Software Keyboard for Smartwatches Using Large Keys to Improve Input Accuracy

Yota Mokuya and Hiroshi Hosobe^[0000–0002–7975–052X]

Faculty of Computer and Information Sciences,
Hosei University, Tokyo, Japan
hosobe@acm.org

Abstract. Smartwatches have smaller screens than smartphones, which results in small keys on their software keyboards. There are mainly three types of Japanese input available for smartwatches, i.e., flick input, Romanized input, and voice input. Romanized input usually uses a QWERTY keyboard that is wide for smartwatches, which causes a problem with aspect ratios. Although voice input does not face such a problem, it cannot be used in environments where speaking aloud is impossible. Therefore, flick input is necessary, but existing keyboards are equipped with twelve flick keys, which makes each key small. This paper proposes a Japanese software keyboard for smartwatches that uses large keys to improve input accuracy. For this purpose, our keyboard reduces the number of keys especially by combining flick keys with tap keys. To evaluate its effectiveness, we conducted an experiment by comparing it with a normal flick input keyboard. The experimental results show that its input accuracy is comparable to that of the normal keyboard.

Keywords: Text entry · Software keyboard · Smartwatch.

1 Introduction

Smartwatches have smaller screens than smartphones, which results in small keys on their software keyboards. Such small keys cause frequent tap errors due to the fat finger problem [5]. There are mainly three types of Japanese input available for smartwatches, i.e., flick input, Romanized input, and voice input. Romanized input usually uses a QWERTY keyboard that is wide for smartwatches, which causes a problem with aspect ratios. Although voice input does not face such a problem, it cannot be used in environments where speaking aloud is impossible. Therefore, flick input is necessary, but existing keyboards are equipped with twelve flick keys, which makes each key small. In addition, flick input is more difficult to use than tap input, which often causes both tap and flick errors.

To solve these problems, previous research has devised various keyboards. Although circular keyboards have been discussed in many papers [1, 6, 7], they are difficult to implement on a rectangular smartwatch, and such a significant change from existing keyboards may affect user experience. Yasufuku and Nakamura [12] developed a Japanese software keyboard called QuadKey that uses

only four keys by adopting two flick operations. The difficulty of operating it may cause users to take time to master it. Jang et al. [3] developed a QWERTY keyboard called UOIT for smartwatches that reduced the number of keys by using a drawing-like typing technique. All alphabets can be entered with tap keys since they are fewer than Japanese kana characters.

In this paper, we propose a Japanese software keyboard for smartwatches that uses large keys to improve input accuracy. We aim to reduce tap errors caused by the small screen of a smartwatch and flick errors experienced by users who are not proficient at flick input. For this purpose, we reduce the number of keys especially by combining flick keys with tap keys and also by modifying the key layout and enlarging the remaining keys. We conducted two experiments to evaluate our keyboard. The first was a preliminary experiment to determine the keyboard layout, and the second was a main experiment comparing our keyboard with a normal keyboard. In the preliminary experiment, we compared three different layouts and identified the best layout in terms of input speed and accuracy. In the main experiment, we compared three variants of our keyboard with a normal flick input keyboard. We associated three different input methods (flick, toggle, and long press) with the keys for selecting vowels in our keyboard. The results showed that our keyboard with flick input was comparable in input accuracy with the normal keyboard.

The rest of this paper is organized as follows. Section 2 presents previous work related to our research, and Section 3 explains Japanese text entry. Next, Section 4 proposes our software keyboard, and Section 5 provides its implementation. Then Section 6 presents the preliminary experiment that we conducted to determine the key layout of our keyboard, and Section 7 gives the main experiment that we conducted to evaluate our keyboard. Section 8 discusses the results of the experiments, and finally Section 9 provides conclusions and future work.

2 Related Work

In Japanese flick keyboards for smartwatches, there are two major solutions to input errors, i.e., enlarging keys and reducing the number of flick keys. Tojo et al. developed a circular Japanese kana character input keyboard called BubbleFlick [7] and its improved version called BubbleSlide [6], and Akamine et al. developed another circular one called PonDeFlick [1]. BubbleFlick allows the entry of different characters by flicking in four directions, whereas BubbleSlide enables the input of characters depending on the flick distance from the center of the screen. PonDeFlick allows the input of characters by first flicking to the center and then flicking in the up, down, left, or right direction. These keyboards use their entire screens, and there is a problem that the entered characters may get hidden by the operating finger.

Yasufuku and Nakamura [12] developed a Japanese input keyboard called QuadKey for smartwatches that uses only four keys. By using two flick operations, it enables the entry of 60 different characters with only four keys. Naka-

mura et al. [4] proposed a Japanese input keyboard called SlicKey that uses five keys and occupies less than 20 % of the smartwatch screen. By using three operations, i.e., touch down, slide, and flick, it enables the entry of 85 characters and commands with five keys. Although these keyboards achieved high accuracy and speed, their complex operations may require time for users to master them.

Toriyama and Miyashita [8] proposed a Japanese kana character input keyboard called InvisibleFlick with transparent key tops. By making the existing flick input keyboard transparent, the background text area becomes visible. However, overlapping keyboard elements still obstruct parts of the text area.

Jang et al. [3] developed a keyboard called UOIT that improves a QWERTY keyboard for smartwatches. The number of its keys was reduced by using the drawing-like typing technique that is adopted in Korean Hangul keyboards. Since alphabets are fewer than Japanese kana characters, all the alphabets can be entered only with tap keys.

Tsunoda and Shizuki [10] proposed a one-handed alphabetical character input method using the accelerometer sensor of a smartwatch. It allows the entry of alphabetical characters by recognizing thumb-to-finger gestures [9] performed by the hand wearing the smartwatch. Although it eliminates the need for an on-screen keyboard and enables one-handed input, limitations remain in sensor recognition accuracy and character input speed.

Wada et al. [11] proposed a Japanese text entry method called Flick-in designed for indirect touch interaction by using a smartwatch as the input surface. Instead of tapping directly on an on-screen keyboard, users perform bezel-initiated swipe gestures on the smartwatch. It separates input on the smartwatch and visual feedback on an external display, which enables text entry on the small screen of the smartwatch.

3 Japanese Text Entry

Basically, Japanese text is written with the mixture of Chinese and kana characters. While a Chinese character is associated with both a meaning and a speech sound, a kana character is associated only with a sound. There are about fifty basic kana characters, and most of them correspond to one consonant and one vowel. For example, the kana character indicating the “ka” sound corresponds to consonant “k” and vowel “a.” For the basic kana characters, the consonants are “k,” “s,” “t,” “n,” “h,” “m,” “y,” “r,” and “w,” and the vowels are “a,” “i,” “u,” “e,” and “o.” There are also five basic kana characters corresponding only to these five vowels (without any constant) and one basic kana character corresponding only to consonant “n” (without any vowel). There are variants of some basic kana characters called dakuon, han-dakuon, and yoh’on. The dakuon variants of consonants “k,” “s,” “t,” and “h” are “g,” “z,” “d,” and “b” respectively, and the han-dakuon variant of consonant “h” is “p.” The yoh’on variants are made by adding “y” to some consonants. For example, the yoh’on variant of “k” is “ky.” A unit of yoh’on is represented as a pair of a kana character and a special kana

character. There is also another special kana character called sokuon that inserts a short pause.

For Japanese text entry on personal computers, a method called Romanized input is popular. It usually adopts standard QWERTY hardware keyboards and lets users enter kana characters by expressing them with alphabets in the way described above.

For software keyboards on smartphones and tablets, many people prefer a method called flick input. A normal flick input-based software keyboard is equipped with ten main and two auxiliary keys. One main key corresponds to the group of the vowel-only basic kana characters, and the other nine main keys correspond to the groups of the basic kana characters that share the same consonants.¹ If one main key is tapped, the corresponding basic kana character with vowel “a” is entered. If one main key is flicked leftward, upward, rightward, or downward, the corresponding basic kana character with vowel “i,” “u,” “e,” or “o” is entered. One auxiliary key is used to specify dakuon, han-dakuon, yoh’on, and sokuon, and the other auxiliary key is used to enter punctuation marks.

4 Proposed Method

In this section, we propose a Japanese software keyboard for smartwatches that reduces the number of keys. Our keyboard supports the entry of Japanese kana characters and determines a kana character in one or two steps. It allows the user to select the consonant of a kana character at the first step, and to select its vowel at the second step.

Our keyboard, shown in Fig. 1, provides the following three kinds of keys.

1. Two keys for consonants (enclosed with a red rectangle in Fig. 1). Each of them is associated with five consonants, and consonants are selected by flick, toggle, or long press input. After a consonant is selected, different characters corresponding to the selected consonant will be shown on the keys for vowels.
2. Four keys for vowels (enclosed with a blue rectangle in Fig. 1). They show kana characters corresponding to the already selected consonant. If one of the four keys is operated by tap input, the corresponding character is selected; if none of the keys is operated, the character corresponding to vowel “a” is selected.
3. Other keys. Most of them are operated by toggle input. An example is to enter punctuation marks, and another is to specify special kana characters and variants of basic kana characters, i.e., yoh’on, sokuon, dakuon, and han-dakuon.

A kind of Romanized input is used by dividing kana characters into consonants and vowels, which allows the user to enter the intended kana character by selecting its consonant and vowel. Flick input is used to select consonants. Since

¹ The consonant-only kana character corresponding to “n” is usually treated as a member of group “w.”

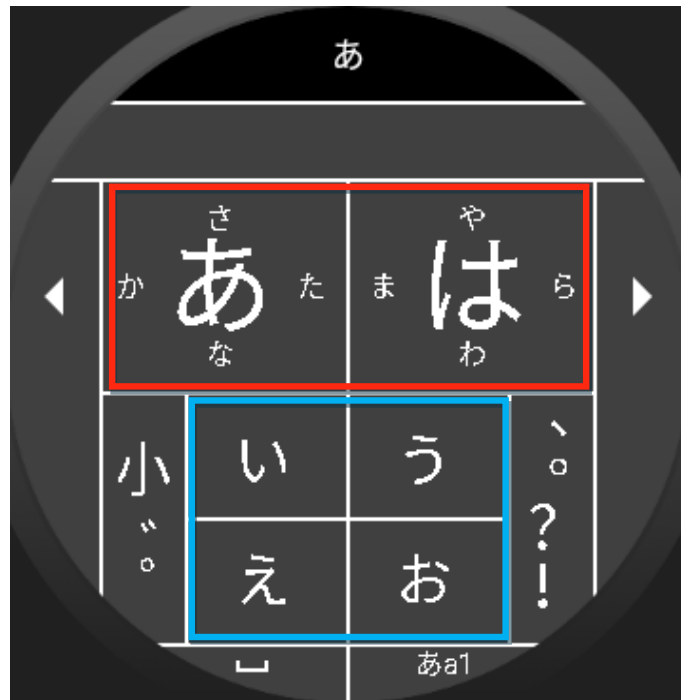


Fig. 1. Proposed Japanese software keyboard for smartwatches. The keys for consonants and the keys for vowels are enclosed with a red and a blue rectangle respectively.

each flick key allows for five different inputs, ten different consonants are identified by two flick keys. Specifically, one key is used for the first five consonants (i.e., vowel-only,² “k,” “s,” “t,” and “n”), and the other key for the second five consonants (i.e., “h,” “m,” “y,” “r,” and “w”). This reduces the number of main flick keys from the typical number of ten to two.

Tap input is used to select four vowels (i.e., “i,” “u,” “e,” and “o”) other than vowel “a.” Also, toggle input is used for punctuation marks (i.e., commas, periods, and question and exclamation marks) and variants of consonants (yoh’on, dakuon, and han-dakuon) since they are less frequently used than basic kana characters. These tap and toggle keys are presented as larger keys than those on typical keyboards. Also, the flick, the tap, and the toggle keys are placed at the top, at the lower center, and at the lower sides respectively, which allows the user to naturally move among these keys while entering characters.

Typical flick input-based keyboards often additionally support toggle input for the same keys. In addition to the flick input described above, our keyboard supports toggle input and also introduces long press input. The newly introduced long press input changes characters by the user’s continuing to touch a key

² For simplicity, we regard the vowel-only kana characters as a special consonant group.

(instead of flicking or toggling on it). A good point with this long press input is that it can be simultaneously implemented with existing flick and toggle input. This long press input changes displayed characters one after another and selects one depending on the timing of releasing the finger. Its main advantage is that keys are pressed once and released once, which reduces operational errors even on small screens. Another advantage is that it does not affect the layout of the keys.

5 Implementation

To evaluate the proposed Japanese software keyboard, we implemented it for a Wear OS-based smartwatch called Mobvoi TicWatch E3. This is a smartwatch of size $44 \times 47 \times 12.6$ mm and is equipped with a 1.3-inch circular screen with a resolution of 360×360 pixels. We developed our software keyboard in Java on Android Studio.

We changed the directions of the flick keys for the main experiment from that for the preliminary experiment. In the preliminary experiment, the four flick directions were suggested horizontally on the top of an operated flick key, which was similar to the flick keys for TicWatch E3's built-in software keyboard. However, many participants in the preliminary experiment commented that the flick keys were difficult to use. Therefore, in the main experiment, we changed the directions of the flick keys to the normal ones (i.e., left, up, right, and down), which is normal for Japanese smartphones.

6 Preliminary Experiment

We conducted a preliminary experiment to determine the appropriate layout of keys for vowels in our software keyboard. For this purpose, we devised three key layouts that we call horizontal, 2-by-2, and flick. The horizontal layout, shown in Fig. 2(a), adopts four tap keys for vowels and places them horizontally, which enables the shortest distance from flick keys for consonants to tap keys for vowels. The 2-by-2 layout, shown in Fig. 2(b), also employs four tap keys for vowels but arranges them in a 2-by-2 grid, which allows the tap keys to be placed in the center. The flick layout, shown in Fig. 2(c), uses another flick key for vowels and places it in the center, which makes it larger than the tap keys for the horizontal and the 2-by-2 layout. In conducting the experiment, we followed the regulations of Research Ethics Involving Human Subjects enacted by our university.

6.1 Procedure

We recruited six participants (four male and two female) for this experiment. Four of them used flick input and the other two used Romanized input in their daily use of smartphones. Only one participant had experience of using a smartwatch. In the experiment, we asked them to wear the smartwatch on one arm

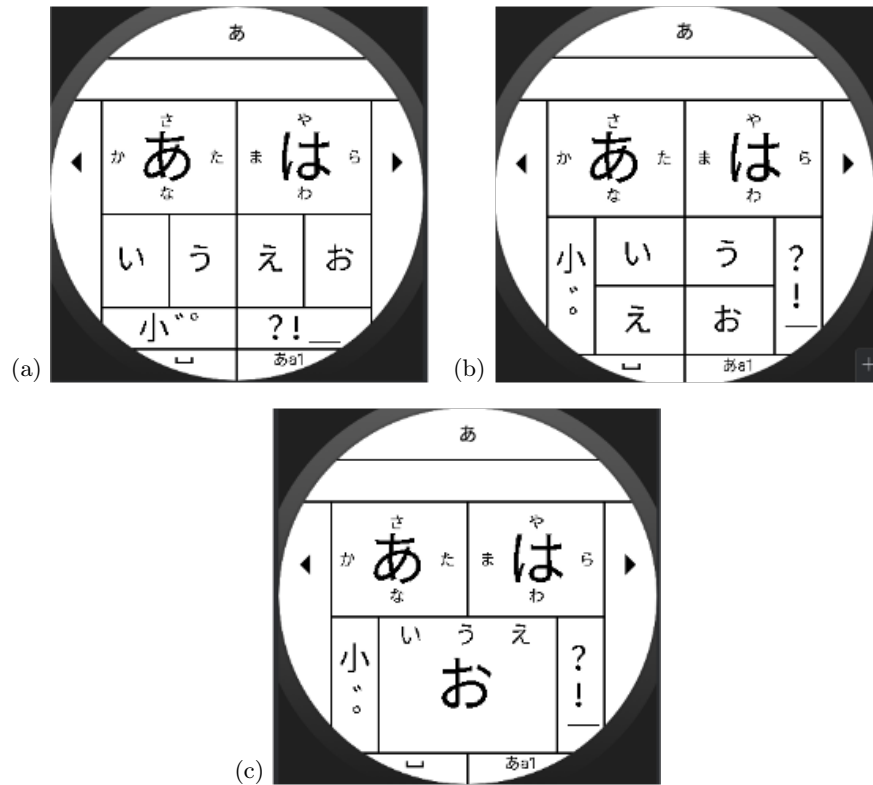


Fig. 2. Key layouts compared in the preliminary experiment: (a) horizontal; (b) 2-by-2; (c) flick.

and enter characters using the other hand. We also instructed them to input as quickly and accurately as possible.

We prepared fifteen short sentences consisting only of kana characters and asked the participants to enter five sentences by using each of the three keyboards. To counterbalance the order effect, we assigned different orders of using the three keyboards to the six participants. We let the participants sufficiently practice the keyboards before the experiment. We measured and compared mean input times per character and mean error rates.

6.2 Results

The results of the experiment are shown in Fig. 3. The mean input times per character were 4.5 seconds for the horizontal layout, 4.4 seconds for the 2-by-2 layout, and 5.3 seconds for the flick layout, with the flick layout using a flick key for vowels being slower than the other two layouts using tap keys for vowels. The mean error rates were 13.0 % for the horizontal layout, 11.9 % for the 2-by-2

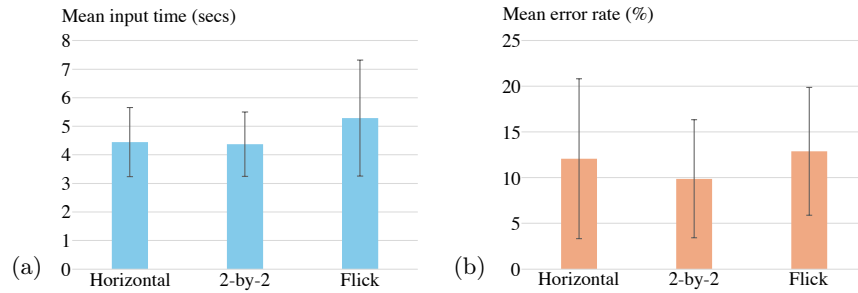


Fig. 3. Results of the preliminary experiment: (a) mean input times per character; (b) mean error rates.

layout, and 12.9 % for the flick layout, with the 2-by-2 layout being the most accurate. Also, the 95 % confidence intervals for the mean input time and the mean error rate of the flick layout were wide.

7 Main Experiment

After the preliminary experiment to determine the appropriate key layout, we conducted a main experiment on the comparison of our final keyboard (shown in Fig. 4(a)) with a normal flick input keyboard with twelve flick keys (shown in Fig. 4(b)). Considering the results of the preliminary experiment, we selected the 2-by-2 layout of the four vowel keys for our final keyboard. Also, we added visual feedback to our keyboard to improve usability and adopted gray keys and a black background to make it more similar to existing smartwatch keyboards. The two consonant keys of our keyboard accept three input methods, i.e., flick, toggle, and long press, which we compared with the normal keyboard in this experiment. We again followed the regulations of Research Ethics Involving Human Subjects enacted by our university.

7.1 Procedure

We recruited eight participants (all male). In the daily use of smartphones, six among them used flick input and the other two used Romanized input. We asked them to wear the smartwatch on the wrist of one hand, to enter text with the other hand, and to try to input as quickly and accurately as possible. Although our keyboard supported three input methods, we asked the participants to use only one of the methods during the experiment.

We prepared twenty short sentences consisting of kana characters and asked the participants to enter five sentences for each of the four compared methods. We assigned the participants with different orders of using the four methods based on a Latin square design. We let the participants sufficiently practice the

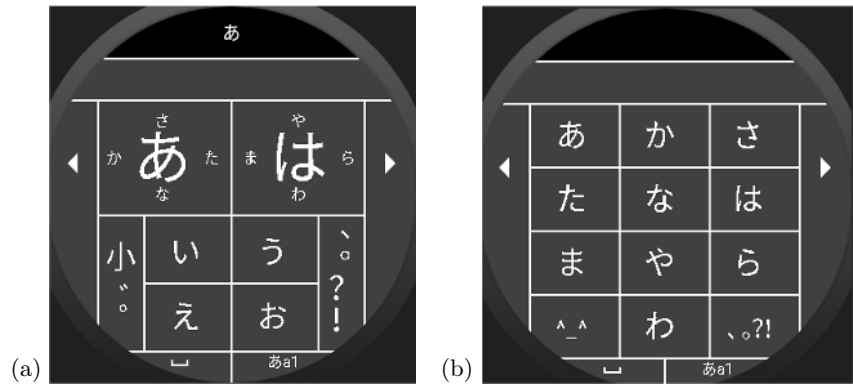


Fig. 4. Software keyboards compared in the main experiment: (a) proposed; (b) normal.

Table 1. Questions of the SUS questionnaire [2].

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

keyboards before the experiment. We measured and compared mean input times per character and mean error rates.

After the experiment, to evaluate the usability of our keyboard, we conducted a questionnaire based on the System Usability Scale (SUS) [2]. The SUS questionnaire consists of ten questions that are listed in Table 1, and each question is rated on five-point Likert scale.

7.2 Results

The results of the experiment are shown in Fig. 5. Notably, it turned out that our keyboard with flick input was comparable in the mean error rates with the normal keyboard. The mean input times per character were 3.8 seconds for our keyboard with flick input, 4.3 seconds for our keyboard with toggle input, 4.3 seconds for our keyboard with long press input, and 2.1 seconds for the normal keyboard, which means that the normal method was faster than our keyboard. The mean error rates were 17.7 % for our keyboard with flick input, 41.3 % for

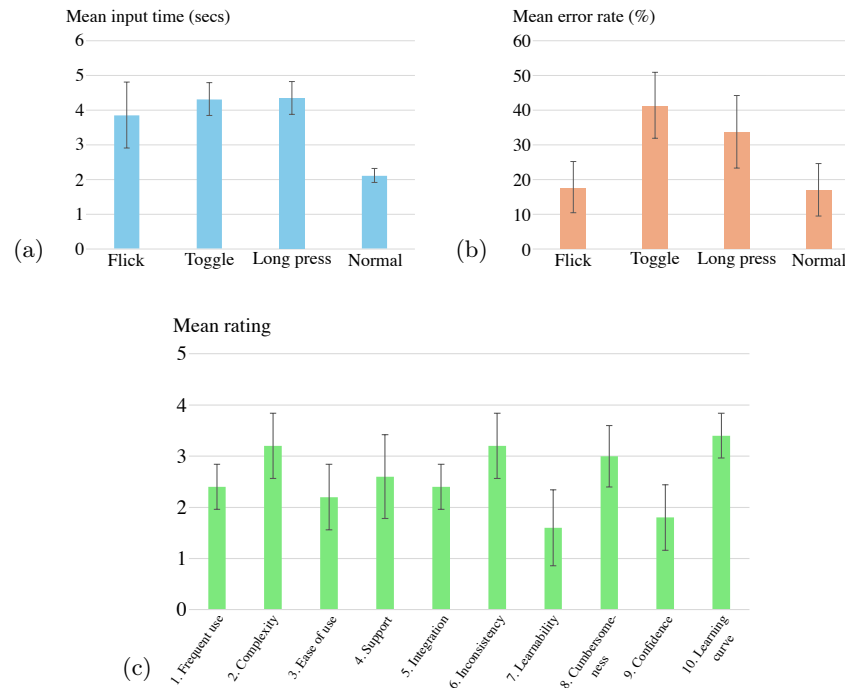


Fig. 5. Results of the main experiment: (a) mean input times per character; (b) mean error rates; (c) mean ratings on the questions of the SUS questionnaire.

our keyboard with toggle input, 33.6 % for our keyboard with long press input, and 17.0 % for the normal keyboard, which means that our keyboard with flick input and the normal keyboard were almost the same in the mean error rates.

The mean SUS score of our keyboard was 64.5, slightly smaller than the normally considered average score of 68. According to the detailed results shown in Fig. 5(c), the participants thought that our keyboard was well integrated, did not need prior support, and was easy to use, but that it took time to understand, and they felt they could not use it with confidence. Some participants commented that it was difficult to enter “sakana” (the Japanese word meaning fish) with one key using toggle input since there were few keys, and that long press input was easier to use than flick input.

8 Discussion

The results of the preliminary experiment suggest that flick input is more difficult to operate than tap input. Tap input requires only pressing and releasing a single key, whereas flick input requires pressing a key, dragging it to another area, and then releasing it. Although the flick key for vowels in Fig. 2(c) was as large as the four tap keys in Fig. 2(b), it produced the highest mean error rate, which

indicates that ease of use is more important than key size. Furthermore, the key layout in Fig. 2(b) outperformed the one in Fig. 2(a) in both input speed and accuracy. Since the tap keys are positioned closer to the center in the key layout in Fig. 2(b) than in Fig. 2(a), this suggests that keys located near the center are easier to use than those placed at the sides.

The results of the main experiment indicate that enlarging keys and reducing the number of flick keys improved input accuracy. Since our keyboard was comparable to the normal one even in the short-term experiment, it is expected that users will make fewer errors once they become fully accustomed to our keyboard. Input speed may also improve as the users become accustomed to our keyboard. However, since typically two operations are required for each character, it is unlikely that our keyboard will much outperform the normal keyboard in terms of input speed. On the other hand, input speed tends to increase with practice, but input accuracy may decrease because the users gradually lose the habit of carefully confirming each character. We need to conduct a long-term experiment to evaluate performance after the users reach proficiency.

In the main experiment, the mean error rate exceeded 40 % when our keyboard with toggle input was used, likely due to differences from the normal keyboard. The toggle input of the normal keyboard changes through vowels “a-i-u-e-o,” whereas the toggle input of our keyboard changes through “a-ka-sa-ta-na” (which changes consonants instead of vowels). Since this behavior was unfamiliar and further operations were required to change vowels, our keyboard with toggle input was more difficult.

Comparing the results of the preliminary and the main experiment for our keyboard with flick input, input speed improved while input accuracy decreased. The primary reason is thought to be the flick directions. The implementation of the flick directions differed between the two experiments: the preliminary experiment required only upward flick operations, which were unfamiliar to the participants, whereas the main experiment used the typical four directions (left, up, right, and down). In the preliminary experiment, the participants needed to check the flick directions, which resulted in slower but more careful input. In the main experiment, the participants no longer needed to check the flick directions, which likely caused more mistakes.

The SUS score of our keyboard was 64.5, slightly below the average score of 68. The participants who normally use Romanized input rather than flick input on smartphones gave particularly low ratings for ease of use and learnability. One possible explanation is that none of the participants had prior experience using a smartwatch and thus they were not accustomed to text entry while wearing a smartwatch. In addition, smartphone operation widely varies among individuals (some use one or both hands, and the choice of fingers differs), whereas text entry on a smartwatch is limited to one hand, with most users operating it using the index finger. Participants unfamiliar with this constraint likely gave lower usability ratings.

As with text entry on smartphones, the problem of the keyboard being hidden by the user’s finger was also observed in our experiments. Although we designed

our keyboard in a top-to-bottom input manner, the fact that text entry begins with the keyboard almost hidden remains an issue for future improvement.

9 Conclusions and Future Work

In this paper, we proposed a Japanese software keyboard that increased key sizes by reducing the number of flick keys. Our preliminary experiment demonstrated that a 2-by-2 tap key layout for vowels yielded the best performance. Then we implemented our keyboard with three input methods for vowels, i.e., flick, toggle, and long press. The main experiment showed that our keyboard with flick input was comparable in input accuracy to the normal keyboard although it remained inferior in input speed.

There are three major challenges for future work. The first is to improve the input speed of our keyboard. One promising approach is to introduce single stroke input. Since our current keyboard requires multiple operations to enter one character, eliminating these operations through the single stroke input will lead to faster input. The second challenge is to reexamine the experimental design. The number of participants in this work was limited, and conducting a larger-scale experiment would likely produce more reliable and unbiased results. Also, since the real-world use of software keyboards typically involves predictive text conversion to omit the latter half of a sentence, further experiments incorporating predictive text conversion will be necessary. The third challenge is to improve the toggle input of our keyboard, for which we can consider two potential solutions. The first is to extend the acceptance time of toggle input. Although a longer acceptance time may hinder smooth input, the end of toggle input in our keyboard automatically transitions to a vowel, which is expected to make the negative impact minimal. The second solution is to introduce visual feedback for the acceptance time of toggle input. Since errors in judging the acceptance time are more common than errors caused by tapping wrong locations, visual feedback could reduce unnecessary input errors.

Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Number JP24K14904.

References

1. Akamine, K., Tsuchida, R., Kato, T., Tamura, A.: PonDeFlick: A Japanese text entry on smartwatch commonalizing flick operation with smartphone interface. In: Proc. ACM CHI. pp. 941:1–11 (2024)
2. Brooke, J.: SUS: A quick and dirty usability scale. In: Usability Evaluation in Industry, pp. 189–194. Taylor and Francis (1996)
3. Jang, R., Jung, C., Mohaisen, D., Lee, K., Nyang, D.: A one-page text entry method optimized for rectangle smartwatches. IEEE Trans. Mob. Comput. **21**(10), 3443–3454 (2022)

4. Nakamura, Y., Yokokawa, R., Honda, T.: SlicKey: A kana character input method for smartwatches using slide and flick operations. *IPSJ J.* **65**(7), 1180–1190 (2024), in Japanese
5. Siek, K.A., Rogers, Y., Connelly, K.H.: Fat finger worries: How older and younger users physically interact with PDAs. In: *Proc. IFIP INTERACT. LNCS*, vol. 3585, pp. 267–280 (2005)
6. Tojo, T., Honda, Y., Kato, T., Yamamoto, S.: BubbleSlide: Annular layout Japanese kana text entry interface for smartwatches. *IPSJ J.* **60**(11), 2075–2084 (2019), in Japanese
7. Tojo, T., Kato, T., Yamamoto, S.: BubbleFlick: Investigating effective interface for Japanese text entry on smartwatches. In: *Proc. ACM MobileHCI*. pp. 44:1–12 (2018)
8. Toriyama, R., Miyashita, H.: InvisibleFlick: A Japanese input keyboard with transparent key tops for a small touch screen device. *IPSJ SIG Tech. Rep.* **2019-HCI-182**(27), 1–8 (2019), in Japanese
9. Tsunoda, R., Choi, M., Shizuki, B.: Thumb-to-finger gesture recognition using COTS smartwatch accelerometers. In: *Proc. MUM*. pp. 184–195. ACM (2024)
10. Tsunoda, R., Shizuki, B.: One-handed character input method based on thumb-to-finger gesture recognition using the accelerometer of a smartwatch. In: *Proc. JSSST WISS*. pp. 106–113 (2023), in Japanese
11. Wada, Y., Choi, M., Shirane, K., Shizuki, B.: Flick-in: Japanese text entry method for indirect touch using bezel-initiated swipe. In: *Proc. ACM SUI*. pp. 19:1–13 (2025)
12. Yasufuku, T., Nakamura, Y.: QuadKey: A Japanese kana character input method for smartwatches with four keys. *IPSJ J.* **60**(8), 1403–1412 (2019), in Japanese