BubbleFlick: Investigating Effective Interface for Japanese Text Entry on Smartwatches

Takaki Tojo
Doshisha University
Kyoto, Japan
ctwb0158@mail4.doshisha.ac.jp

Tsuneo Kato
Doshisha University
Kyoto, Japan
tsukato@mail.doshisha.ac.jp

Seiichi Yamamoto
Doshisha University
Kyoto, Japan
seyamamo@mail.doshisha.ac.jp

ABSTRACT

We propose BubbleFlick, an effective interface for Japanese text entry on smartwatches. While various ideas have been proposed to provide easy and fast text entry for the Latin alphabet, Japanese text entry has additional challenges such as having more than fifty syllabary characters, or kana, to enter and the subsequent kana-kanji conversion, which translates a sequence of the syllabary characters into a standard expression with a mixture of kanji and kana characters. This paper focuses on interfaces for entry of kana syllabary characters. We designed and prototyped three interfaces: 1) Japanese kana syllabary keyboard, 2) Dial&Flick interface, and 3) DualBelts interface. Through a comparative pilot study of the prototypes, we refined the most promising Dial&Flick interface into BubbleFlick. BubbleFlick provides the widest possible area for easy flick operations while also leaving an area for editing text. We conducted a 30-day consecutive user study on BubbleFlick in comparison with Google's latest Japanese text-entry method based on a numeric keypad. After thirty days, BubbleFlick showed a text-entry speed of over 35 characters per minute, which was comparable to Google's numeric-keypad-based method for novice participants. Through the user study, BubbleFlick showed a lower error rate and gave us informative hints for further improvement.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces: Input devices and strategies

Author Keywords

BubbleFlick; text entry; software keyboard; Japanese kana; smartwatch

INTRODUCTION

Smartwatches are expected to be the next leading type of wearable devices. Providing an easy and quick text-entry method other than automatic speech recognition is demanded for enriching interactivity and accelerating prevalence.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobileHCI '18, September 3–6, 2018, Barcelona, Spain

© 2018 ACM. ISBN 978-1-4503-5898-9/18/09...\$15.00

DOI: <https://doi.org/10.1145/3229434.3229455>

The main challenge of text entry on smartwatches is simply to achieve easy and fast operability on a small touchscreen. There have been various attempts to help users control tiny keys by zooming in a specific area of a keyboard [22], by splitting a software keyboard into multiple pages [11] and by displaying the touched area in a non-occluded area [17]. There have been proposals of assigning multiple characters to each of a reduced number of keys and estimating a word with disambiguation techniques [18, 15, 4, 7, 21]. Or one of the assigned characters is specified by flick operations [24] or by finger identification [8]. There have also been gesture-based text entries for words [16, 30, 31] and for characters [27].

Meanwhile, recent statistical decoding techniques have achieved accurate key-entry detection based on probabilistic touch models [5, 9, 28, 6]. A keyboard based on a statistical decoder incorporating a probabilistic touch model, a character N-gram model, and a word language model recorded a text-entry speed as fast as 41 words per minute (WPM) [28].

However, most such proposals have been developed or verified on miniature QWERTY keyboards, although many of them can be applied to other keyboards. Text entry of Japanese has different characteristics from that of western languages. Because the Japanese language is written using thousands of kanji characters, text is first entered with kana, i.e., syllabary characters, and then converted to a standard Japanese text style by invoking a kana-kanji converter or a predictive converter. Since the kana, either hiragana or katakana, has more than fifty characters, the kana entry needs more efficient interface than Latin alphabet entry does. Furthermore, one-to-many correspondence between a kana sequence to texts in the standard Japanese style needs an effective predictive kana-kanji converter such as POBox [20], which suggests a selectable list of probable words after every input,

Japanese text entry on mobile devices has been developed in a unique way. Since short message service (SMS) and mobile internet service had been popular on feature phones before the prevalence of smartphones, Japanese text entry on a numeric keypad had already been established. In the text entry, five kana characters comprised of a specific consonant and either one of five vowels were assigned to one numeric key, and the desired kana character was specified by multiple taps. This interface was taken over to smartphones, and the multi-tap interface incorporated flick operations which select one of the five kana characters by either using four flick directions or simply tapping the key [29]. This is a kind of a menu-

augmented soft keyboard [14] based on a numeric keypad. The sequence of kana entered is then processed by the predictive kana-kanji converter to show a list of probable words.

Yet, simple adoption of the standard numeric keypad in a smartwatch makes both the key size and the spacing between keys very small. Hence, we focus on an effective interface for Japanese kana text entry, leaving the kana-kanji conversion aside in this paper. Through a comparative pilot study of three interfaces based on different design concepts, we propose BubbleFlick, which is based on the concept of maximizing the usability of flicking by providing the widest area possible. BubbleFlick arranges twelve buttons corresponding to the twelve keys of a numeric keypad, circularly at the perimeter of a watch face so that the flick operations use the entire touchscreen.

We conducted a 30-day user study with eight participants to see how users learn a new interface on such a small touchscreen. The participants entered specified texts every day using two methods: BubbleFlick and Google's latest Japanese text entry based on a numeric keypad. The user study revealed how the text-entry speed and error rate changed with time, and provided a hint for further improvement.

RELATED STUDIES

Zooming a part of a keyboard has been a popular approach in smartwatch text entry. ZoomBoard [22] works in a two-step operation. It zooms in on a small keyboard iteratively, and the desired key is specified by an additional tap. The callout type of keyboard displays the character about to be entered in a non-occluded area and lets the user change the character if needed by shifting the touching finger slightly. ZShift keyboard [17] improved upon the callout by displaying a zoomed-in image of the touched area in a non-occluded area. This study compared the three methods described above on three different screen sizes (16 mm, 24 mm and 32 mm wide) and concluded that the preferred method depends on the screen size. SplitBoard [11] displays a half of a QWERTY keyboard on a small touchscreen and lets a user switch to the other half. SwipeBoard [1] lets a user swipe twice to enter a character; the first swipe specifies the area where the character is located, and the second swipe specifies the desired character.

On the other hand, fast-growing statistical decoding techniques have permitted precise detection of key touches and accurate prediction of next words. Google's WatchWriter [6] provided tap typing and gesture typing on a miniature QWERTY keyboard. Google built this system upon their Smart Touch Keyboard and Smart Gesture Keyboard techniques developed on smartphones [23]. The paper reported text entry speeds of over 20 WPM with character error rates (CER) of less than 4 % for both typing modes. VelociTap [28] achieved very high accuracy with a sentence-based decoder incorporating a probabilistic touch model, a 12-gram character language model and a 4-gram word language model. That paper reported 41 WPM on a 40-mm-wide keyboard with a 3% CER.

No matter how statistical decoding provides precise text entry, big keys with the proper density in a small touchscreen ease text entry while reducing the need for excessive carefulness. In

a context similar to Japanese text entry, four popular key layouts for Korean text entry were evaluated on a smartwatch [13]. Three layouts based on a numeric keypad were taken over to smartphones from feature phones [12], and one QWERTY-like Korean keyboard was added as a baseline. That paper concluded that the numeric-keyboard-based layouts are viable with a competitive text-entry speed and a reasonable CER. However, Korean text entry is based on multi-tap operation. We consider that users need a wider area for flick operations. As a Japanese text entry method, No-look flick was proposed for eyes-free text entry based on three buttons; two for consonants and one for vowels, with the four-directional flick operations on a handheld device[3]. Assigning multiple characters to each of a reduced number of keys has also been proposed for Latin alphabet keyboards. An ambiguous keyboard estimates sequences of characters of probable words from a set of ambiguous keys [15]. The work on SwipeKey [24] investigated the relation between the error rate and the number of flick directions per key in various key layouts on a smartwatch. DualKey [8] assigns two characters to a key, and the system selects either character by identifying whether the index finger or the middle finger tapped the key. Our BubbleFlick is rather a variant of a menu-augmented soft keyboard [14] based on a ringed key layout. The menu-augmented soft keyboard was extended to multiple character entry by KeyScratch [2].

The ringed key layouts appeared in earlier work. Cirrin [19] arranged 26 alphabetic characters in a ringed pattern for gesture-based text entry that supports continuous strokes for selecting consecutive characters with a stylus. BubbleCIRCLE [10] allowed both tap typing and gesture strokes on a ringed arrangement of alphabetic characters with an additional visual effect of magnifying the probable next characters. BubbleCIRCLE was proposed for an interactive tabletop display.

THREE PROTOTYPE CHARACTER-ENTRY INTERFACES FOR JAPANESE KANA SYLLABARY

We designed and prototyped three types of Japanese kana-entry interfaces based on different design concepts. Note that the following prototypes were designed under a strong restriction of using no rightward flick operation, which was assigned to termination of the application at the time of prototyping.

Japanese Kana Syllabary Keyboard

Japanese kana is a syllabary character. One kana character corresponds to one Japanese syllable, which is basically composed of a consonant and a vowel (CV), or only a vowel (V). Japanese sounds have five vowels, 'a', 'i', 'u', 'e', 'o', and nine basic consonants. A total of 46 basic kana characters, which comprise the five vowels, forty CV syllables and a nasal sound 'n,' are allocated on a Japanese kana syllabary table that has 5 rows and 10 columns. The rows correspond to the five vowels, and the columns correspond to the occurrence of just a vowel without a consonant and the basic nine consonants from right to left. (In practice, additional 25 syllables with voiced consonants, 33 syllables with contracted sounds, and a syllabary character representing doubled consonants are used along with the basic characters.) The Japanese kana syllabary table is well-known to Japanese speakers, from children to elderly people.

Figure 1. Japanese kana syllabary keyboard.

Figure 2. Dial&Flick interface.

Figure 3. DualBelts interface.

—	わ	ら	や	ま	は	な	た	さ	か	あ	—
、		り	ゆ	み	ひ	に	ち	し	き	い	ゝ
。	を	る	よ	む	ふ	ぬ	つ	す	く	う	。
ゝ		れ	や	め	へ	ね	て	せ	け	え	、
。	ん	ろ	ゆ	も	ほ	の	と	そ	こ	お	。
—	、	。	よ	—	。	ゝ	っ	—	。	、	ゝ

Figure 4. Extended kana syllabary table. Four additional kana characters for contracted sounds and doubled consonants are arranged in the table. Three symbols for long syllables and punctuation marks, and two symbols for voiced consonants, are lined up in the left, right, and bottom around the original table of 5 rows and 10 columns.

Figure 5. Screenshot of Dial&Flick interface when a button is selected. When a finger touches one of the buttons on the ring, the circular area inside the ring is partitioned into four radial areas for four kana characters of the column.

The first interface, a Japanese kana syllabary keyboard, is shown at an actual scale in Figure 1. This prototype is based on the concept of intuitive understandability. The keyboard area is displayed in the lower half of the watch face, and the text-editing area is displayed on top of the keyboard area. Tapping a key of the keyboard based on the square Japanese kana syllabary table is intuitively understandable to every Japanese speaker. This interface considers only ease of visual search without consideration of travel distance. Each key has a size of 6.2 mm square. As the size of the keyboard exceeds that of the watch face, a user drags the partly displayed keyboard to the right position if needed, and taps a kana key. Two buttons, one for shifting the keyboard to the right end and the other for resetting the keyboard at the initial position, are added on the left end of the watch face, since a rightward flick operation is not allowed. The two buttons shown on top of the screen are used for “enter” and “delete.”

The entire image of the Japanese kana syllabary keyboard is shown in Figure 4. It is an extended table of 6 rows and 12 columns. Four additional kana characters are arranged properly in the table, and a repetitive series of three symbols and two marks are lined up in the left, right and bottom around the original table of 5 rows and 10 columns.

Dial&Flick Interface

The second interface, Dial&Flick, is designed based on the concept of maximizing the usability of flick interface for smart-watches. As stated in the Introduction, flick operation for

text entry is nowadays very popular in Japan because it is widespread on smartphones. The concept is got into shape by providing as wide an area as possible for flicking and leaving a wide enough area for editing text at the same time. The initial screen of the Dial&Flick interface is shown in Figure 2.

The principle of how to specify a kana character is based on two steps: first specifying a consonant and then second a vowel. Here, ten representative kana characters, which are located in the first row of the kana syllabary table and two buttons for symbols and marks, which make up twelve in total, are arranged in a ring at the perimeter of a watch face. The representative kana characters are laid out in the order of the kana syllabary keyboard in the clockwise direction starting from a 3 o'clock position. Because Japanese speakers are used to the standard kana order in the kana syllabary table, the representative kana characters are arranged in the order, rather than considering travel distance. The text-editing area is located in the center of the ring. A user starts tracing the ring like dialing from the yellow button at the 3 o'clock position to one of the buttons, makes a turn, and flicks to a kana character. Figure 5 shows a screenshot when a user's finger touches one of the twelve buttons, in this case 'a.' When the user's finger touches one of the representative kana buttons, the circular area inside the ring is partitioned into four radial areas for flicking, and the other four kana characters with the same consonant are displayed in the quartered areas in addition to the representative kana button. The size of a button is 9 mm in the perimeter direction and 5 mm in the perpendicular

Figure 6. Relative frequency distributions of kana characters for sentence sets.

direction. The text-editing area is 20 mm in height and 24 mm in width. The order of the four characters in the quartered areas are in the anticlockwise direction. The kana character to enter is determined by the released position. Hence, the representative kana character is entered when the user releases the finger on the button. A kana character in a quartered area is entered when the user flicks to one of the areas. The reason for starting the flick from the yellow button is to circumvent a rightward flick. The determination of whether a flick is rightward or not is based on the relative position of the released point from the landed point in the horizontal direction.

DualBelts Interface

The third interface, DualBelts, is designed based on the concept of minimizing the area required for a kana-entry interface and leaving as wide an area as possible for the text-editing area. Its screen is shown in Figure 3.

This interface is also based on the principle that a kana character is a combination of a consonant and a vowel. The vertical belt is for specifying a column of the kana syllabary table, while the horizontal belt is for specifying a row, that is, determining a kana character. The vertical belt has eleven keys of the representative kana characters in the order of the kana syllabary table. The size of a key is 6 mm square. The vertical belt is placed on the right side for a right-handed user. A wide open area for editing text is located in the upper left part. A user selects a column of the kana character by sliding the vertical belt at first, and the characters displayed on the horizontal belt change depending on the selected column. Then, the user taps a kana character on the horizontal belt. The horizontal belt, which contains from five to ten keys, can be slid if necessary. Since a sliding operation takes priority to a rightward flick on the horizontal belt, the interface is not terminated by a rightward sliding operation.

PILOT STUDY

Evaluation Method

Participants

A pilot study comparing the three prototypes was conducted with twenty participants. The participants were twelve male and eight female persons of ages ranging from 19 to 53. Nineteen of them were right-handed and the other one was left-handed. All except one owned a smartphone, and nobody owned a smartwatch. Year of their smartphone usage ranged from 0 to 7 years.

Apparatus

The smartwatch used was the Motorola Moto 360. It has a round face with a diameter of 39.6 mm and a resolution of 205 ppi.

Text for Entry

The text for the pilot study was a set of ten Japanese short sentences. One sentence consisted of from seven to twenty kana characters, but without a kanji character in order to measure only time for kana entry. The sentences were kept simpler and shorter than typical sentences people write so that the participants would not be confused while text entry. One sentence contained 2.6 content words with 12.1 kana characters in average. Kana character balance was taken into account. Figure 6 shows the relative frequency distribution of kana characters of this sentence set, with that of the sentence sets for the user study, and that of a corpus of 15k sentences wrote on a smartphone-based Japanese SNS. The distribution of this sentence set was correlated with that of the 15k-sentence corpus. The correlation coefficient was 0.70.

Procedure

The evaluation was conducted at a table in a quiet room of our laboratory. Participants received a briefing and wore the smartwatch on their non-dominant wrist and operated it with the index finger of the dominant hand. They learned how to operate each of the interfaces by entering a set of sentences for practice at first. Then, the participant entered the predefined set of 10 short sentences with each of the interfaces. The time required for completing a sentence and the number of delete key entries were measured by the kana-entry application sentence-by-sentence. The order of testing the three interfaces was set randomly across the participants to avoid an order effect. Participants were asked to assess the usability of each prototype on a 7-point Likert scale (7 = excellent) along with giving a free description in a post-evaluation questionnaire.

Objective Measures

Performance was measured by text-entry speed in character per minute (CPM) and total kana error rate, which counts both corrected errors and uncorrected errors [26] in terms of kana characters, in error per character (EPC).

Evaluation Results

Figure 7 shows mean CPMs with standard deviations between participants. The mean CPMs were 16.2, 17.0 and 11.8 for Japanese kana syllabary keyboard, Dial&Flick, and DualBelts,

Figure 7. Mean text-entry speeds of three prototypes in character per minute (CPM).

Figure 8. Mean total kana error rates of three prototypes in error per character.

respectively. The standard deviations between participants were 4.0, 4.3 and 2.8, respectively. There was not significant difference between Japanese kana syllabary keyboard and Dial&Flick with a risk rate of 5% in paired t-test.

Figure 8 shows total kana error rates. The mean total kana error rates were 13.1%, 7.9% and 12.2% for Japanese kana syllabary keyboard, Dial&Flick, and DualBelts, respectively. The standard deviations were 8.0, 6.5 and 7.3. Though the standard deviations were generally large due to individual variability, paired t-tests showed significant differences between Dial&Flick and the other two with a risk rate of 5%.

Figure 9 shows mean opinion scores (MOS) with standard deviations. The mean scores were 3.5 for Japanese kana syllabary keyboard, 5.0 for Dial&Flick, and 2.9 for DualBelts.

At the end of the pilot study, the major comments from the questionnaire's free description are listed below.

1. Japanese kana syllabary keyboard

- This design was the easiest to use among the three designs because of its visibility.
- The button for resetting the keyboard at the initial position was useful after learning how to use it.
- It was inconvenient that a rightward flick was not allowed.
- It might be better to add a button for shifting to the left as well as shifting to the right.

Figure 9. Mean opinion scores (MOS) of three prototypes in 7-point Likert scale.

- My finger became cramped due to the small touch area.

2. Dial&Flick interface

- The flick operation like on a smartphone was easy to use.
- The quartered areas for flicking were hidden behind my finger when entering a kana character in the upper-left area.
- I forgot the text that I was editing while text entry.
- I wanted to start flicking from any button on the ring.

3. DualBelts interface

- It was nice that kana characters with voiced consonants were listed on the belt.
- I wanted the horizontal belt to slide faster.
- It might be better to set the belts as fixed.
- It is not easy to stop the sliding belt at an intended position. I was frustrated when the intended kana character did not appear.
- The fonts were too small.

Comparison of Three Prototypes

In terms of CPM, the Japanese kana syllabary keyboard and Dial&Flick interfaces were more than 1.4 times faster than the DualBelts interface. Two-step operations for entering every kana character with the DualBelts interface took more time than the other two interfaces, which allow the user to enter a kana character with a single-step operation.

In terms of the total kana error rate, the Dial&Flick interface showed the lowest level. In the Japanese kana syllabary keyboard, the relatively small buttons might have caused more errors. In the DualBelts interface, the greater number of operations required for entering a kana character might have caused more errors. In the subjective assessment, Dial&Flick got the highest rating.

Based on the pilot study's results, we decided to refine the Dial&Flick interface. The Dial&Flick got the highest score among the prototypes, and it still had a great room for further leaps in CPM and in usability. If we could resolve the issue of the long stroke starting from the yellow button at the right end, the flick operation could be simplified much more and CPM would be expected to significantly increase.

Figure 10. Screenshots of BubbleFlick interface.

DEVELOPMENT OF BUBBLEFLICK

We developed BubbleFlick as an extension of the Dial&Flick interface. The initial screen and two snapshots when a finger is on a button are shown in Figure 10. The initial screen is not very different from the Dial&Flick interface. Twelve buttons are lined up in a ring at the perimeter of a round watch face in the clockwise direction from a 3 o'clock position, and the text-editing area is located in the center. However, a user can start flicking from any button in the ring, rather than dialing from a button at a 3 o'clock position. This is achieved by allowing rightward flick operation in BubbleFlick. When a user touches one of the representative kana buttons, the other four kana characters of the same consonant are displayed like bubbles around the touched button. The order of four characters around the touched button is basically in the clockwise direction except for the buttons located at 9 o'clock to 12 o'clock positions. The anticlockwise direction for the buttons located at 9 o'clock to 12 o'clock was decided based on a number of comments at a preliminary test. In addition, a square callout pops up to show the kana character currently selected. The kana character to enter is determined by the position of the releasing point.

BubbleFlick puts the first priority on the ease of visual search, but it cuts the travel distance greatly. Major modifications from Dial&Flick are listed below.

1. Starting flicking from any character in the ring is permitted by overriding the terminating command of a rightward flick. This modification provides simple operability and improves efficiency of text entry by eliminating the dialing operation required in the Dial&Flick interface.
2. The size of reacting area of a button is 7 mm in diameter with a kana character of 4 mm square. The spacing between two neighboring kana characters is set at 4 mm.
3. A 9mm-square callout of the selected kana character is displayed away from the touched area to prevent occlusion.
4. "Enter" and "delete" buttons are removed from the screen by assigning the functions to certain flick operations on the text editing area, instead. Specifically, an upward flick operation off any button is assigned to "enter," while a downward flick operation off any button is assigned to "delete." Furthermore, left and right flick operations off any button are assigned to "move cursor left / right", respectively.

Figure 11. Google's Japanese text-entry method on a smartwatch.

USER STUDY

Evaluation Method

We conducted a user study lasting 30 days to clarify how participants learn a new text-entry interface in such a small touchscreen. While we were developing BubbleFlick, Google released a new Japanese text-entry method that allows flick and multi-tap operations in a small numeric keypad on a smartwatch. Google's Japanese text entry method for smartwatches was a natural extension of the most popular Japanese IME on smartphones. Thus, we conducted a user study comparing BubbleFlick with Google's keypad-based flick and multi-tap text-entry method with eight participants. We call Google's product KeypadFlick for short hereafter. We asked the participants to continue using two kana-entry methods for 30 days straight to see how they got used to using them.

Participants

The participants were seven males and one female of ages ranging from 22 to 25. None of them was a participant of the pilot study. Seven out of the eight were right-handed. All of them owned a smartphone, but none owned a smartwatch. Years of their smartphone usage ranged from 4 to 8 years. Five participants usually entered text with a numeric keypad on their smartphones, four of them used both flick and multi-tap operations on the keypad, and one used only multi-tap operation on the keypad. The other three participants usually entered text with a QWERTY keyboard. We categorized the four participants who usually used flick and multi-tap operations on a numeric keypad of their smartphones as KeypadFlick User Group and the other four who usually used multi-tap operations on a numeric keypad and a QWERTY keyboard of their smartphones as KeypadFlick Novice Group.

Figure 12. Daily text-entry speeds (CPMs) of eight participants in using BubbleFlick.

Figure 13. Daily text-entry speeds (CPMs) of eight participants in using flick kana-entry on numeric keypad (KeypadFlick).

Apparatus

The model of the smartwatch used was Motorola Moto 360, 2nd Generation. The Moto 360 2nd Gen has a slightly smaller watch face, with a diameter of 34.8 mm, than the original Moto 360, which has a watch face with a diameter of 39.6 mm. The watch face has a resolution of 263 ppi.

Text for Entry

We prepared 30 sets of Japanese short sentences to present the participants a new set day-by-day. We kept each set of sentences in a small amount because people would not enter so much text with such a small text-entry interface everyday. Each set was composed of from five to six sentences with basic words, which amounted to about 100 kana characters in total and formed a kana pangram as a whole. Pangram means text that includes all letters at least once. One sentence contained 3.5 content words with 16.6 kana characters in average. The relative frequency distribution of kana characters is shown in Figure 6. The distribution was correlated with that of the corpus of a smartphone-based Japanese SNS, but was generally smoothed because they were pangram. The correlation coefficient was 0.74.

Each set of sentences was printed on a sheet with a date of usage. The date of usage was not randomized across the participants. A regular expression with a mixture of kanji and kana characters was printed for each kana sentence because it is not always easy to enter text with only kana sentences. The sets of sentences can be evaluated in terms of a difficulty level with difficulty levels of words. Among tens of thousands existing kanji characters, a list of 2136 are designated as kanji characters for common use by the Cabinet in Japan. The sets of sentences contained 465 kanji characters, and 448 (i.e. 96%) of them were included in the list of kanji characters for

common use. This shows the sentences are easy enough to evaluate text-entry performance.

Procedure

Every participant was asked to enter a specified set of sentences everyday using two kana-entry methods: BubbleFlick and KeypadFlick. The participants were instructed to perform daily text entry in a quiet environment without a contact from others and at a set hour as best as they could. They were strictly instructed to wear the smartwatch on their non-dominant wrist and operate it with the index finger of the dominant hand in a fixed posture that they initially found best-suited to themselves. The order of using the two methods was alternating day-by-day, and printed on every sheet of the sentence set. The participants were asked to read through the sentence set of the day before starting the daily text entry not to be confused in reading them while text entry. On the first day of the evaluation, the participants were instructed the daily procedure and asked to enter a set of sentences as a practice to understand how to operate the smartwatch and the kana entry methods. We let them know that all of the history of text entry, specifically the flicked positions, entered characters, and time information, would be recorded in the smartwatch. After the second day, they entered a set of sentences at home everyday. They were not allowed to enter additional text with the smartwatch for private use. The experimenter met the participants and collected the recorded data every week. They were asked to answer a questionnaire after 30 days.

Text-entry Method Based on Numeric Keypad

The text-entry method compared with BubbleFlick was Google's latest Japanese text-entry method for the Android smartwatch. The initial screen and a screenshot when a button

participant	BubbleFlick				KeypadFlick			
	y-intercept	slope	RMSE	corr.	y-intercept	slope	RMSE	corr.
#1	31.32	0.59	3.36	0.83	34.00	0.69	5.20	0.75
#2	22.75	0.34	2.12	0.81	36.49	0.43	4.76	0.62
#3	24.89	0.47	3.15	0.79	33.74	0.55	5.36	0.66
#4	23.31	0.45	3.34	0.75	32.10	0.53	5.85	0.60
#5	22.06	0.31	2.79	0.69	15.77	0.58	3.76	0.80
#6	22.91	0.22	3.08	0.54	24.07	0.42	6.05	0.52
#7	26.68	0.73	3.73	0.86	23.23	0.65	3.21	0.87
#8	17.32	0.47	2.36	0.86	18.98	0.50	4.25	0.71

Table 1. Y-intercept and slope of regression line approximating text-entry speed (CPM) of each participant. Root mean square error and correlation coefficient are attached.

is touched are shown in Figure 11. The software keyboard has a 3x4 keypad with 10 representative kana characters in the first row of the Japanese kana syllabary table. It is a miniature version of the popular Japanese kana-entry method on smartphones, which is based on flick operations on a numeric keypad. The upper half of the watch face is the text-editing area, and the lower half is the text-entry area. When a key is touched, a flick guide is displayed over the numeric keypad, showing five kana characters on flick directions. Seven keys surrounding the lower side of the keypad are used to 1) move the cursor leftward, 2) switch character types, 3) enter symbols, 4) space, 5) delete, 6) enter and 7) move the cursor rightward, from left to right. Furthermore, predicted word candidates are displayed over the keypad. The size of a key is 3 mm in height and 7 mm in width.

Results

Text-entry Speed

Figure 12 and Figure 13 show text-entry speeds (CPMs) of the eight participants for 30 days straight with BubbleFlick and KeypadFlick, respectively. Each bar graph shows daily CPM of an individual participant. Missing bars indicate missing data due to the participants' forgetfulness. Participants from #1 to #4 are in the KeypadFlick User Group, who usually enter text with KeypadFlick on their smartphones. Participants from #5 to #8 are in the KeypadFlick Novice Group. Specifically, participants from #5 to #7 had never used KeypadFlick on their smartphones, and participant #8 had used only multi-tap entry with a numeric keypad. A moving average over 7 days and a regression line are drawn with an orange line and a dashed line, respectively. Every graph shows an upward trend with daily variability. Most of the bar graphs show that the upward trend slowed down after around day 20, but did not reach a plateau clearly except for #4, #5 and #8. The y-intercept, which represents the estimated initial speed, and the slope, which represents the estimated learning rate, of the regression line for each individual participant are listed with the root mean square error (RMSE) and correlation coefficient (corr.) in Table 1.

There was an apparent gap in CPMs between the KeypadFlick User Group and the KeypadFlick Novice Group for KeypadFlick, whereas no gap was observed between the groups for BubbleFlick. This is an effect of habituation of the text-entry method on their smartphones. Figure 14 shows daily CPMs averaged over three groups, all participants, KeypadFlick User

Group and KeypadFlick Novice Group. Two curves of BubbleFlick for two sub-groups are omitted because they mostly overlap that of BubbleFlick for all participants. Table 2 shows mean estimated CPMs of the groups for every (roughly) ten days based on the regression lines.

Before comparing groups, let us compare the CPMs for the Dial&Flick and BubbleFlick. It is a less unfair comparison between the mean CPM for the Dial&Flick in Figure 7 and the mean CPM of all participants for BubbleFlick on the first day. Though BubbleFlick was tested on a slightly smaller watch face, BubbleFlick recorded a speed of 1.4 times faster in CPM (24.4) compared to the Dial&Flick (17.0).

Next, let us compare the CPMs for BubbleFlick and KeypadFlick for each group. The KeypadFlick User Group showed clearly faster entry speeds in KeypadFlick than in BubbleFlick. Meanwhile, the KeypadFlick Novice Group showed comparable speeds, but with a little steeper learning rate in KeypadFlick. We conducted ANOVA test with the text-entry methods and periods being within-subject factors and the user groups being a between-subject factor. The results showed a significant interaction of the text-entry methods and user groups ($F_{(1,6)} = 10.40, p < .05$). The sub-effect test showed a significant simple main effect of the text-entry methods in KeypadFlick User Group ($F_{(1,6)} = 20.67$), but did not in KeypadFlick Novice Group ($F_{(1,6)} = 0.00$). Regarding the slope of the regression lines, ANOVA test did not show a significant main effect of the text-entry methods ($F_{(1,6)} = 5.82, p < .10$).

Error Rate

Figure 15 and Figure 16 show total kana error rates, which count both corrected and uncorrected errors[26], of the eight participants for 30 days using BubbleFlick and KeypadFlick, respectively.

Generally, BubbleFlick showed significantly lower error rates than did KeypadFlick. The mean total kana error rates of eight participants were 7% for BubbleFlick and 14% for KeypadFlick. ANOVA test showed a significant main effect of the text-entry methods ($F_{(1,6)} = 31.33, p < .01$). The sub-effect test showed significant simple main effects of the text-entry methods both in KeypadFlick User Group ($F_{(1,6)} = 22.96$) and KeypadFlick Novice Group ($F_{(1,6)} = 9.75$). The total kana error rates for BubbleFlick were as low as 10% from the first day, while those for KeypadFlick showed a decreasing trend for most of the participants. However, the mean total kana

Figure 14. Daily text-entry speeds (CPMs) Averaged over All Participants, KeypadFlick User Group and KeypadFlick Novice Group.

	BubbleFlick				KeypadFlick			
	day 1	day 10	day 20	day 30	day 1	day 10	day 20	day 30
All participants	24.4	28.4	32.8	37.3	27.8	32.7	38.1	43.6
KeypadFlick User Group	26.0	30.2	34.8	39.5	34.6	39.6	45.1	50.6
KeypadFlick Novice Group	22.7	26.6	30.9	35.2	21.1	25.8	31.2	36.5

Table 2. Estimated text-entry speeds (CPMs) of BubbleFlick and KeypadFlick based on regression lines. The regression lines are averaged over three groups: all participants, KeypadFlick User Group, and KeypadFlick Novice Group.

error rate for BubbleFlick (14% on the first day) was greater than that for Dial&Flick interface (8%).

Total kana error rates were dependent on the individual. The error rate of participant #7 was significantly lower than the others. On the other hand, that of participant #5 was greater than the average, and his CPM for KeypadFlick was the slowest among the eight participants. Participant #5 had a large hand. His large hand seemed to cause frequent errors, especially in using KeypadFlick, and this is known as the “fat finger” problem [25]. In total kana error rates for KeypadFlick, no clear gap was observed between the KeypadFlick User Group and the KeypadFlick Novice Group.

Interview Research

The experimenter interviewed the participants on the usability of two kana-entry methods after the 30-day assessment. The major pros and cons collected from the comments are listed below.

1. BubbleFlick

[Pros]

- It has a good design. Functions were easily comprehensible.
- I made few entry errors with the help of the large fonts.
- I made few entry errors with the help of the great spacing between buttons.
- Errors that I made were noticeable because of the visibility of the selected kana character and entered text.
- Deleting operation of entered text was easy.

[Cons]

- It was hard for me to memorize flick directions because the flick direction of the same vowel changes depending on the button, i.e. the representative kana.
- I sometimes made errors in the directions of flick on the text-editing area in the center.

- It was troublesome to move a cursor character-by-character by horizontal flick operations in the text-editing area.

2. KeypadFlick

[Pros]

- It was easy to use because it is the interface that I am using on my smartphone everyday.
- The flick directions were easy to memorize because of the systematicity.
- It was easy to use because the area for touching was limited to the lower half of the watch face.
- The delete button was useful.
- Errors that I made were noticeable by referring to the list of predicted words.

[Cons]

- The keypad was hard for me to operate because of the highly packed area for touching.
- It was troublesome to enter the same kana character successively.
- The text-editing area was narrow and hard to see.
- I sometimes could not operate the cursor as I wished to.

BubbleFlick got many favorable comments on its visual design and usability. Specifically, the circular button layout and its intuitive operability were favorably received. Larger fonts, greater spacing between buttons, and a wider text-editing area earned the system a favorable reputation as intended.

On the other hand, unsatisfactory comments on BubbleFlick were mainly on its difficult-to-remember flick directions, which depend on the button the user starts flicking from. KeypadFlick has systematic flick directions that correspond to the five vowels, specifically, releasing the finger on the button for

Figure 15. Daily total kana error rates of eight participants using BubbleFlick.

Figure 16. Daily total kana error rates of eight participants using flick kana entry on a numeric keypad (KeypadFlick).

‘a,’ flicking leftward for ‘i,’ upward for ‘u,’ rightward for ‘e’ and downward for ‘o’. In contrast, BubbleFlick has different flick directions depending on the button from which a user starts flicking because these buttons are located at the perimeter of the watch. This issue is treated again as a remaining issue in Discussion.

DISCUSSION

BubbleFlick provided about 1.4-times-faster CPM (24.4) on the first day of use compared to that of Dial&Flick (17.0). This leap was due to the simplified operability of BubbleFlick.

CPMs showed a continuing upward trend with time, although daily variability was observed. On the other hand, the total kana error rates did not show a clear decreasing trend for BubbleFlick. We consider that this is due to the less carefulness of the participants because a character can be entered more quickly with easy operability. In the case of KeypadFlick, total kana error rates over 15% decreased to a certain level, but did not keep decreasing monotonously. We consider that this is simply due to the small keys.

We compared BubbleFlick with KeypadFlick. For the KeypadFlick User Group, KeypadFlick showed a significantly faster CPM than BubbleFlick did even after 30 days. For the KeypadFlick Novice Group, the CPMs of BubbleFlick and KeypadFlick were at a comparable level, although BubbleFlick showed a lower CER than KeypadFlick did. The main problem of BubbleFlick is the difficult-to-remember flick directions. As stated before, the flick direction to specify a kana character changes depending on which button the user starts flicking from. Seven out of eight participants commented that the flick directions of BubbleFlick were hard to memorize, and they could not master BubbleFlick even after 30 days.

Our next work is to introduce systematic flick directions into BubbleFlick.

Limitation

To cancel the influence of individual variability on CPM and CER with a limited number of participants, this user study adopted evaluation of two text-entry methods in parallel with the same participant. Though there was no comment related to confusion between two methods, there was a possibility of asymmetric skill transfer between the methods. Specifically, the learning rate might be slowed down due to the influence of learning another method. For example, CPM of KeypadFlick for the KeypadFlick User Group might be slowed down by learning BubbleFlick. We should conduct a larger scale user study with more participants with a separation that a participant assesses either one of the methods.

CONCLUSIONS

We proposed an effective interface for Japanese text entry on smartwatches based on a design concept of providing as wide an area as possible for easy flick operations. BubbleFlick opens up the entire watch face to flick operations and leaves a wide enough area for text editing at the same time. The user study in comparison with KeypadFlick showed a comparable text-entry speed of over 35 CPM after 30 days for KeypadFlick novice participants. The mean CER of BubbleFlick was 7%, which was half that of KeypadFlick.

Moreover, the user study revealed interesting facts. 1) The advantage of familiarity with KeypadFlick on smartphones persisted even after 30-day use. 2) An important factor still missing for easy and fast flick operations in BubbleFlick is a systematic and intuitive arrangement of flick directions.

REFERENCES

1. X. A. Chen, T. Grossman, and G. Fitzmaurice. 2014. SwipeBoard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 615–620. <http://doi.acm.org/10.1145/2642918.2647354>
2. G. Costagliola, V.0 Fuccella, and M. Di Capua. 2011. Text Entry with KeyScretch. In *Proceedings of Intelligent User Interface (IUI '11)*. ACM, New York, NY, USA, 277–286. <http://doi.acm.org/10.1145/1943403.1943445>
3. Y. Fukatsu, B. Shizuki, and J. Tanaka. 2013. No-look flick: single-handed and eyes-free japanese text input system on touch screens of mobile devices. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 161–170. <http://doi.acm.org/10.1145/2493190.2493243>
4. M. Goldstein, R. Book, G. Alsio, and S. Tessa. 1999. Non-keyboard QWERTY Touch Typing: A Portable Input Interface for the Mobile User. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 32–39. <http://doi.acm.org/10.1145/302979.302984>
5. J. Goodman, G. Venolia, K. Steury, and C Parker. 2002. Language Modeling for Soft Keyboards. In *Proceedings of Intelligent User Interface (IUI '02)*. ACM, New York, NY, USA, 194–195. <http://doi.acm.org/10.1145/510000.502753>
6. M. Gordon, T. Ouyang, and S. Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3817–3821. <http://doi.acm.org/10.1145/2858036.2858242>
7. N. Green, J. Kruger, C. Faldu, and R. St Amant. 2004. Non-keyboard QWERTY Touch Typing: A Portable Input Interface for the Mobile User. In *Proceedings of CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 1429–1432. <http://doi.acm.org/10.1145/985921.986082>
8. A. Gupta and R. Balakrishnan. 2016. DualKey: Miature Screen Text Entry via Finger Identification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 59–70. <http://doi.acm.org/10.1145/2858036.2858052>
9. N. Henze, E. Rukzio, and S. Boll. 2012. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2659–2668. <http://doi.acm.org/10.1145/2210000.2208658>
10. U. Hinrichs, H. Schmidt, T. Isenberg, M. Hancock, and S. Carpendale. 2008. BubbleType: Enabling Text Entry within a Walk-up Tabletop Installation. In *Report 2008–893–06*. Department of Computer Science, University of Calgary.
11. J. Hong, S. Heo, P. Isokoshi, and G. Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1233–1236. <http://doi.acm.org/10.1145/2702123.2702273>
12. I. Ilinkin and S. Kim. 2010. Evaluation of Text Entry Methods for Korean Mobile Phones, a User Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2023–2026. <http://doi.acm.org/10.1145/1753326.1753633>
13. I. Ilinkin and S. Kim. 2017. Evaluation of Korean Text Entry Methods for Smartwatches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 722–726. <http://doi.acm.org/10.1145/3025453.3025657>
14. P. Isokoski. 2004. Performance of Menu-Augmented Soft Keyboards. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 423–430. <http://doi.acm.org/10.1145/985692.985746>
15. A. Komninos and M. Dunlop. 2014. Text Input on a Smart Watch. *IEEE Pervasive Computing* 13, 4 (2014), 50–58.
16. P. O. Kristensson and S. Zhai. 2004. SHARK 2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. <http://doi.acm.org/10.1145/1029632.1029640>
17. L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 669–678. <http://doi.acm.org/10.1145/27022123.2702388>
18. F. Chun Yat Li, R. T. Guy, K. Yatani, and K. N. Truong. 2011. The 1Line Keyboard: A QWERTY Layout in a Single Line. In *Proceedings of the Annual ACM Conference on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 461–470. <http://doi.acm.org/10.1145/2047196.2047257>
19. J. Mankoff and G. D. Abowd. 1998. Cirrin: A Word-Level Unistroke Keyboard for Pen Input. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST 1998)*. ACM, New York, NY, USA, 213–214. <http://doi.acm.org/10.1145/290000.288611>

20. T. Masui. 1999. POBox: An Efficient Text Input Method for Handheld and Ubiquitous Computers. In *Proceedings of International Symposium on Handheld and Ubiquitous Computing (HUC 1999)*. Springer, Berlin, Germany, 289–300.
https://doi.org/10.1007/3-540-48157-5_27
21. Nuance. 2016. T9. (2016). <https://www.nuance.com/mobile/mobile-solutions/text-input-solutions/t9.html>.
22. S. Oney, C. Harrison, A. Ogan, and J. Wiese. 2013. ZoomBoard: A Diminutive QWERTY Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802.
<http://doi.acm.org/10.1145/2858036.2858242>
23. S. Reyat, S. Zhai, and P. O. Kristensson. 2015. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 679–688.
<http://doi.acm.org/10.1145/2702123.2702597>
24. Y. F. Shao, M. C. Ogimoto, R. Pointner, Y. C. Ling, C. T. Wu, and M. Chen. 2016. SwipeKey: A Swipe-based Keyboard Design for Smartwatches. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 60–71.
<http://doi.acm.org/10.1145/2935334.2935336>
25. K. A. Siek, Y. Rogers, and K. H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction (INTERACT05 '05)*. Springer, Verlag Berlin, Heidelberg, 267–280.
<http://doi.acm.org/10.1145/2858036.2858242>
26. R. W. Soukoreff and I. S. MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 113–120.
<http://doi.acm.org/10.1145/642611.642632>
27. K. Ueno, K. Go, and Y. Kinoshita. 2016. Design and Evaluation of EdgeWrite Alphabets for Round-Face Smartwatches. In *Proceedings of the Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 177–178.
<http://doi.acm.org/10.1145/2984751.2984757>
28. K. Vertanen, H. Memmi, J. Emge, S. Reyat, and P. O. Kristensson. 2015. Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 659–668.
<http://doi.acm.org/10.1145/2702123.2702135>
29. Wikipedia. 2016. Japanese Input Method. (2016).
https://en.wikipedia.org/wiki/Japanese_input_methods.
30. S. Zhai and P. O. Kristensson. 2012. The Word-Gesture Keyboard: Reimagining Keyboard Interaction. *Communication ACM* 55, 9 (2012), 91–101.
31. S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan. 2009. Shapewriter on the iPhone: From the Laboratory to the Real World. In *Extended Abstracts on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 2667–2670. <http://doi.acm.org/10.1145/1520340.1520380>