Meyboard: A QWERTY-based Soft Keyboard for Touch-typing on Tablets

Yuki Kuno and Buntarou Shizuki

University of Tsukuba, Japan
{kuno,shizuki}@iplab.cs.tsukuba.ac.jp

Abstract. In this paper, we present a soft keyboard, called Meyboard, which enables the user to touch-type. To achieve this, we designed the Meyboard such that the bottom row of the QWERTY layout is omitted, and the positions and sizes of the keys are automatically adjusted to user's fingers while the user is typing. Omitted keys are input by performing a flicking action on the remaining keys. This design allows the user to type faster and more accurately than with a conventional soft keyboard, even if the user does not see the keys while typing. We have implemented a Meyboard prototype, and have performed a longitudinal user study involving one participant. The results show that in the final week of the user study, the participant is able to type at a rate of 40.7 words per minute, with an error rate of 4.57%.

Keywords: Touch Screen · Text Entry · Eyes-free Input · Adaptable Interface · Gestures · Flick · Reduced Keyboard · Longitude User Study

1 Introduction

Touch-typing is the typing method in which a user does not need to visually observe the positions of the keys, but rather detects the positions of keys by touching them while typing. This results in a highly effective typing performance, because the user can focus on the text without shifting his/her eye line to detect the positions of keys. In contrast, conventional soft keyboards displayed on the touch screens of tablet computers (referred to hereafter as tablets) are difficult to use for touch-typing. The reason for this is that the user cannot feel the positions of keys by touching them, because the keys are merely images displayed on the flat surface of the touch screen. Thus, the user is forced to adjust the positions of his/her fingers to the keys by seeing both of them while typing.

The aim of our research is to develop a soft keyboard for tablets with a touch screen that facilitates touch-typing. The key idea is to place keys at the positions of each of the user's fingers. To achieve this, we implemented a soft keyboard that identifies the user's fingers. This enables the user to input text more accurately than on an conventional soft keyboard, even if the user cannot feel the keys. We implemented a prototype, called Meyboard (Fig. 1), which adopts a QWERTY-based layout.

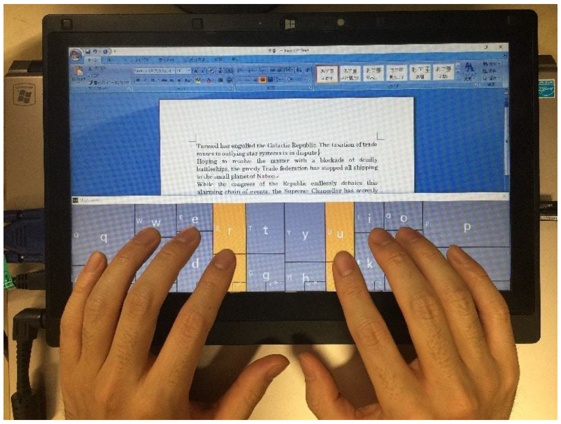


Fig. 1. Typing with the Meyboard.

2 Related Work

A similar approach to our research, where the shapes and positions of keys are adjusted to the user's hands, has been adopted by LiquidKeyboard [12], CATKey [5], personalized input [3], the study of Gunawardana et al. [7], and the study of Go et al. [6]. In contrast to the above work, we focus on improving the performance of touch-typing on a soft keyboard by dynamically adjusting the QWERTY-based layout.

Research on input for visually impaired users has achieved nonvisual input on soft keyboards. BrailleType [11], BrailleTouch [13], and Perkinput [1] adopt Braille, where the user inputs a letter by inputting the Braille pattern corresponding to the letter. While these soft keyboards suit users who are familiar with Braille, we posit that a different method is required for users who do not satisfy this assumption. Our approach exploits the user's familiarity with the QWERTY layout.

While the Meyboard adopts flicking actions for typing, the adoption of flicks on a QWERTY-based soft keyboard is not new. Gestyboard 2.0 [2] also uses flicks to type keys and achieve touch-typing using the QWERTY layout. Although Gestyboard uses flicks for typing most of the keys, in order to maintain the stability of the input, we consider flicking to be stable but time-consuming. Therefore, we designed the Meyboard to employ flicks as seldom as possible in typing.

We omitted the bottom row of the QWERTY layout, in order to reduce the number of keys in the Meyboard and achieve touch-typing. The 1Line keyboard [8] adopts a similar approach. This is a QWERTY-based soft keyboard with only one row. Therefore, the input from the user contains ambiguities. The input is determined by selecting a word from a list that is estimated by the system using the dictionary. In contrast, our approach contains neither a phase for word selection nor a dictionary. Instead, we attempt to achieve touch-typing by doubling the number of keys compared with the 1Line keyboard, and adopting additional input actions.

3 Meyboard

The layout of the Meyboard (Fig. 2) is based on the QWERTY layout, with some additional features.

- The position and size of each key is adjusted to fit the user's fingers while the user is typing.
- The bottom row of the QWERTY layout (i.e., that containing Z–M and some punctuation keys) is omitted. We hypothesized that the user could precisely touch keys without seeing them if we limit the number of keys to one or two in each column. We still display the omitted keys in a "collapsed" format, for practice.

In addition, we changed the colors of the R, F, U, and J keys to make it easier for the user to visually locate these keys while practicing with the Meyboard. We also displayed the letter that will be typed when holding the shift key on left side of each key.

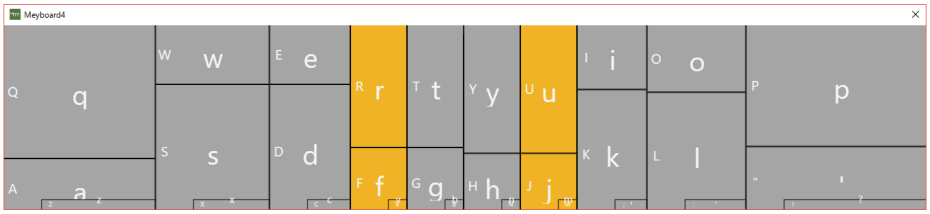


Fig. 2. Key layout of the Meyboard.

3.1 Adjusting Keys (Calibration)

When the user places four fingers (index to little fingers) on the touch screen, the Meyboard defines the key layout for one half of the keyboard. For example, if the user places their four left fingers down, then the top and middle rows of the left half of the QWERTY layout (i.e., the positions and sizes of Q–T and A–G) are defined. We refer to this action as *calibration.* To describe the process roughly, the heights of the keys are defined using the coordinates of the touch points of the user's four fingers, and the widths of the keys are defined using the distancing between those coordinates. Note that if the user place eight fingers (index to little fingers of both hands), the Meyboard defines the key layout for both halves of the keyboard.

The details of the calibration are as follows.

- The height of keys in each column in the top row are set as the distance between the top of the Meyboard and the position of the each finger, and the height of keys in each column in the middle row (in the QWERTY layout) are set as the distance between the position of the each finger and the bottom of the Meyboard, if the following process of fixing the height is ignored. Therefore, every height of keys in a row are different with each other (Fig. 3). Prior to the above process, adjustments are made to these y coordinates to prevent the keys from being too small to touch effectively. Specifically, all of the touch points are moved downwards so that the midpoint of the vertical distance between the highest and the lowest touch points is on the center line of the Meyboard (Fig. 4). As a result, the heights of the keys are made sufficiently large, while keeping them adjusted to the user's fingers.
- The width of a key is set as the horizontal distance between the two midpoints of three adjacent fingers, as shown in Fig. 3. The exceptions to this are the R, F, U, and J keys, where the widths of these are set as twice the horizontal distance between the index finger and the midpoint of the index and middle fingers, to ensure that the T, G, Y, and H keys are reachable for the index fingers. The midpoint between the left and right index fingers is considered as the boundary between the left and right halves of the keyboard. Therefore, the widths of T and G (Y and H) are set as half of the horizontal distance between the edge of R and F and the edge of U and J.

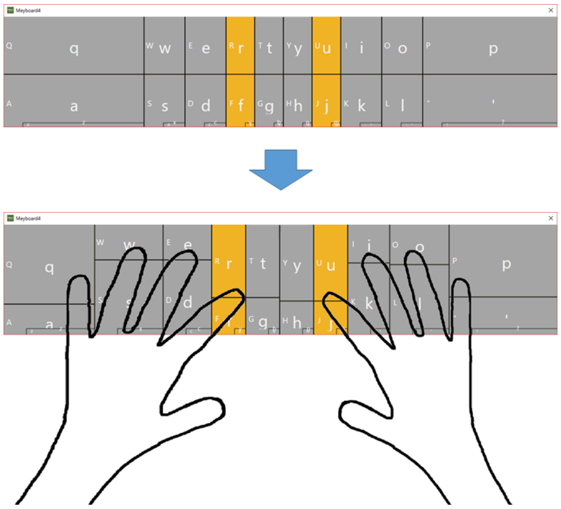


Fig. 3. Calibration of the Meyboard.

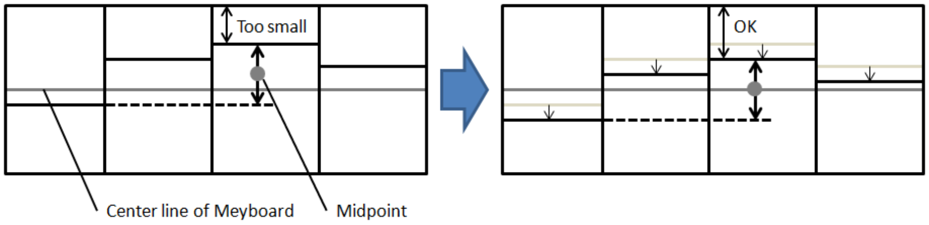


Fig. 4. Height adjustment of keys in the Meyboard.

3.2 Input Method

To allow the user to input 30 letters (alphabet + punctuations) with 20 keys, we divided keys into several areas, as follows, and assigned a specific input action to each area, as shown in Fig. 5:

Home (A–F, J–L) and UpperHome (Q–R, U–P, A–F, J–L, and quote)
These are the home keys and keys above the home keys. Home keys are those where the user places their fingers in the default position for touch-typing on a hard keyboard.

UnderHome (Z–V, M, comma, period, and question)
These are the keys placed below the home keys.

TopCenter and MiddleCenter (T, Y, G, and H)
These are the keys placed at the center of the top and middle rows of the QWERTY layout.

BottomCenter (B and N)
These are keys placed at the center of the bottom row of the QWERTY layout.

Note that the some punctuation keys in this layout (semi-colon and slash) are replaced with others that have higher rates of use (quote and question), according to the layout of the standard soft keyboard in Windows 10.

Before describing each input action, we will define the tap and flick actions that are used as the fundamental input actions in the Meyboard. If the user touches down on the touch screen with their finger and removes it within 450 milliseconds without moving the finger more than 2.8 mm, then this is identified as a tap by the Meyboard. On the other hand, if the user touches down on the touch screen with their finger and removes it within 500 milliseconds after moving the finger more than 2.8 mm, then this is identified as a flick by the Meyboard. These thresholds are determined empirically.

Home and UpperHome
Tapping of keys is used in these areas. We assigned the fastest fundamental input action in the Meyboard to the keys here, because these keys are considered to be used more frequently than those in other areas.

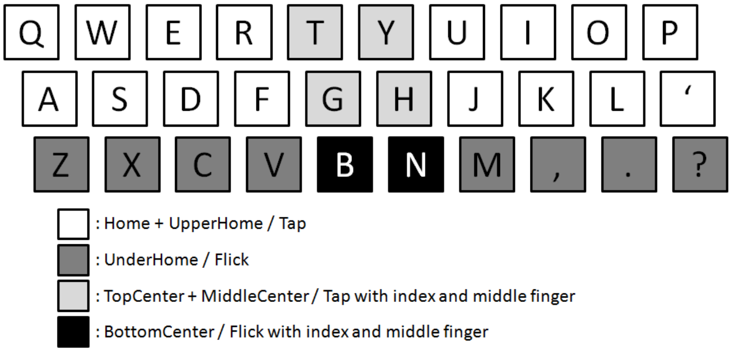

Q W E R T Y U I O P
A S D F G H J K L ‘
Z X C V B N M , . ?
: Home + UpperHome / Tap
: UnderHome / Flick
: TopCenter + MiddleCenter / Tap with index and middle finger
: BottomCenter / Flick with index and middle finger

Fig. 5. Areas that divide the keys of the QWERTY layout, and input actions for typing these keys in the Meyboard.

UnderHome

The actions of flicking keys down is used here. For example, if the user flicks downwards on either the Q or A key, a Z is entered. Although flicking is slower than tapping, its input action is simple, and can easily be learned by the user. Moreover, the keys in the UnderHome area are considered to be used less frequently than those in the Home and UpperHome area. Therefore, the drawback that flicking is slow is diminished compared to its merits.

From this point on, we simply refer to a downwards flick as a flick, because only downwards flicks are used in the Meyboard.

TopCenter and MiddleCenter

Here, the tapping of keys with both the index and middle fingers simultaneously is used. Typing is distinguished between the TopCenter and MiddleCenter by the row of the key that is tapped by the index finger. If the user taps a key in the top row of the QWERTY layout (R or U) with their index finger, then the key from TopCenter is entered (T or Y). If the user taps a key in the middle row of the QWERTY layout (F or J) with their index finger, then the key from MiddleCenter is entered (G or H). In a conventional soft keyboard, when the user wants to type keys from the TopCenter or MiddleCenter area, they must move their finger horizontally to tap these keys. This results in losing the horizontal position of the keys. In contrast, in our approach the user does not move their finger horizontally to type these keys, which solves the above problem.

BottomCenter

The flicking of keys with both the index and middle fingers simultaneously is used here. This input action can be considered as a combination of the inputs for the UnderHome area and for the TopCenter and MiddleCenter areas. We chose this action because BottomCenter shares the same row as UnderHome and the same column as TopCenter and MiddleCenter in the QWERTY layout. Therefore, we postulated that a combination of the inputs of UnderHome and TopCenter or MiddleCenter could be easily learned by the user. In the same manner as when typing keys from the TopCenter and MiddleCenter area, only the movement of the index finger is used to determine if the input is a flick.

We designed the Meyboard to perform the calibration (i.e., setting the key layout for one half of the keyboard) automatically while the user is typing. To achieve this, Meyboard performs the calibration when the user inputs keys that are frequently used in sentences, namely space, enter, shift, and backspace, where we assigned four or eight finger touch gestures (index to little fingers of one or both hands) to these keys. Specifically, space is entered by tapping the left four fingers; enter is entered by tapping the right four fingers; shift is entered by placing and holding the four fingers of either the left or right hand, and the input continues until the four fingers are removed; and backspace is entered by tapping the four fingers of the right hand while placing and holding the four fingers of the left hand. In particular, the user has to input backspace by using eight fingers. As a result, this design adjusts all of the keys of the Meyboard to the latest position of the user's fingers whenever the user tries to correct an error, thus preventing further error.

4 User Study

We conducted a user study to investigate the touch-typing performance of the Meyboard. Fig. 6 illustrates our experimental setup, where we used a Panasonic Let's Note CF-AX2 (11.6 inches, 1366 x 768 pixels) as the tablet. We did not display any keys on the Meyboard in this study, only the area of the board itself (Fig. 7), in order to simulate touch-typing conditions. The input area of the Meyboard is 1366 x 350 pixels, which is approximately 10.4 inches. The participant was one of the authors. This user study was conducted using a customized application whose original was a typing practice application released freely [14].

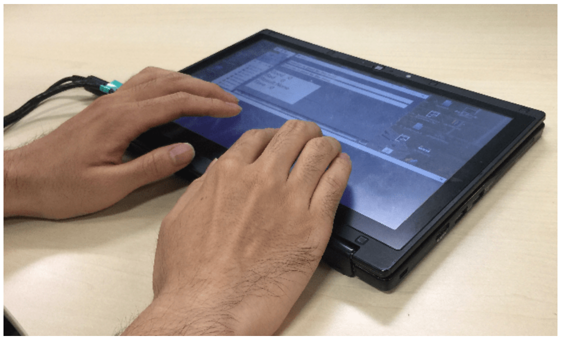


Fig. 6. Experimental setup.

Fig. 7. Meyboard in the user study, where no keys are shown.

We chose the task of inputting short English phrases. The phrases are chosen from the phrase set provided by MacKenzie et al. [9], but the first word of each phrase is capitalized, and a period follows the last word. We capitalized the first letter with the intention of activating the calibration of the Meyboard at least once during the input of a phrase, by forcing the input of a shift, which activates the calibration. In this study, the participant always has to correct any errors they make. If the participant makes any errors, then a corresponding number of cross marks appear. Such a cross mark disappears if the user inputs a backspace to correct the error. Note that inputting a backspace also activates the calibration.

The participant input three sets each day, each consisting of 10 different phrases. The phrases in each set were chosen randomly, without duplicates. Inputting the three sets took approximately 10 to 15 minutes. The study was conducted for 147 days. Therefore, 441 sets were typed by the participant, which is equivalent to 1,323 phrases or 136,207 letters (not counting errors).

4.1 Results

Fig. 8 shows the input rate in words per minute (wpm) [4], with a fitted logarithmic curve. This is calculated as follows:

$$\frac{1}{5}\frac{\text{Incoming keystrokes except mistakes (times/set) - 1}}{\text{Input time (minutes/set)}}$$

The maximum wpm achieved using the Meyboard in the user study was 45.6. The mean and standard deviation of the input rate in the final week of the study were 40.7 wpm and 2.67 wpm, respectively.

Fig. 9 shows the error rate. The minimum error rate achieved using the Meyboard in the user study was 0.32%. The mean and standard deviation of the error rate in the final week of the study were 4.57% and 1.80%, respectively.

When a long-term study is conducted, the observed performance tends to follow a power law [10]. As a result, the curve of the performance becomes almost straight in a double logarithmic chart. Fig. 10 shows the double logarithmic chart of the time required to input 100 characters. As the curve is nearly straight, the performance observed in this study can be considered reasonable. Moreover, it is predicted that the input rate would reach approximately 47.6 wpm after one year (i.e., after 1095 sets).

We compared the touch-typing input rate achieved using the Meyboard with results from related work. Li et al. [8] showed that the average input rate for a soft keyboard on a 9.7 inch iPad (with the QWERTY layout) and on a 1Line keyboard were 53.9 wpm and 30.7 wpm, respectively. Note that the participants in this evaluation were able to type on a hard keyboard at 82.6 wpm, which is faster than most casual users. Therefore, the performance of the participants also tended to be faster on the iPad soft keyboard. Considering the above, the touch-typing input rate of 40.7 wpm achieved on the Meyboard is noteworthy. Gestyboard [2], which is a the QWERTY-based keyboard employing flicks, has achieved an input rate of 21.6 wpm after typing 11,000 characters. Considering the time that passed until the total number of input characters reached that number in our Meyboard study, the input rate estimated using the fitted line shown in Fig. 10 is 27.2 wpm, which indicates the potential of the Meyboard. The average input rate of the Braille-based soft keyboard Perkinput [1] was 38.0 wpm, which is approximately the same as the Meyboard. Overall, these results suggest that the touch-typing performance achieved using the Meyboard in this study is promising.

5 Discussion

To analyze the content of the participant's errors, we collected logs of all key inputs during the study, and counted the number of letters in each phrase to be typed and the numbers of errors in (Fig. 11).

Fig. 11a shows the number of letters to be typed during the user study. This result shows that letters from the bottom row make the fewest appearances. (Capital letters: top 2,544, middle 1,480, and bottom 707. Small letters: top 57,209, middle 33,684, and bottom 20,193) This result is consistent with the design of the Meyboard, where we omitted the bottom row of the QWERTY layout and instead employed a flick action, which is easy to learn but time consuming. Fig. 11b shows the numbers of errors for each letter. Fig. 11c shows the error ratio for each letter, which is the number of errors (Fig. 11b) divided by the number of appearances (Fig. 11a) multiplied by 100 (to give a percentage). Among the capital letters, which require a shift to type, Z, Q, G, A, B, and R are prone to errors. Among the small letters, z, j, x, w, b, and c are prone to errors. These results indicate that inputs with the left hand are more prone to errors, and this tendency increases with inputs using the shift key. In particular, inputs that use a flick with the left little finger, i.e., z or Z, show the highest error ratios.

To further analyze the types of error the participant had made for these letters, we identified every incorrect or omitted input using the following algorithm:

1. The algorithm compares the phrases in the study and the participant's input in terms of letters. Correct inputs are skipped.

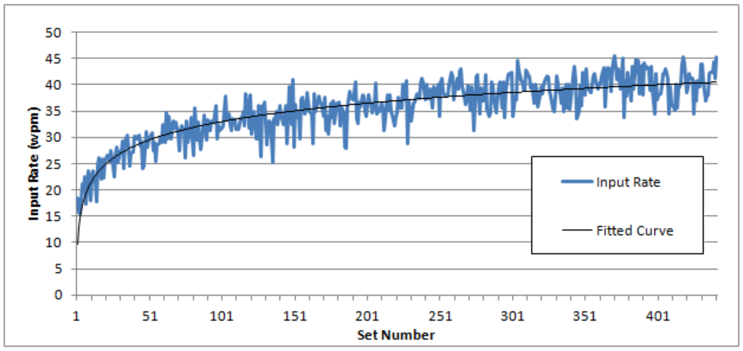


Fig. 8. Input rate (wpm) on the Meyboard.

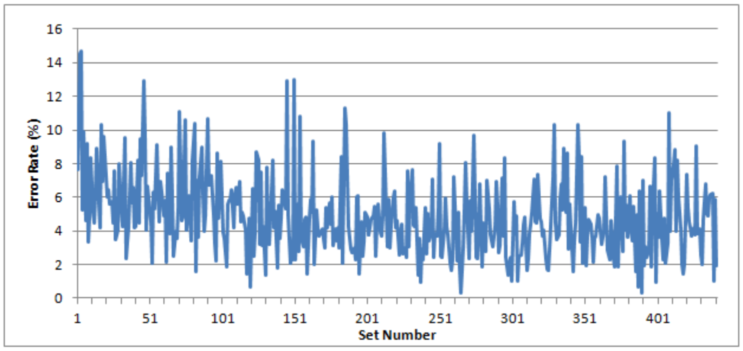


Fig. 9. Error rate on the Meyboard.

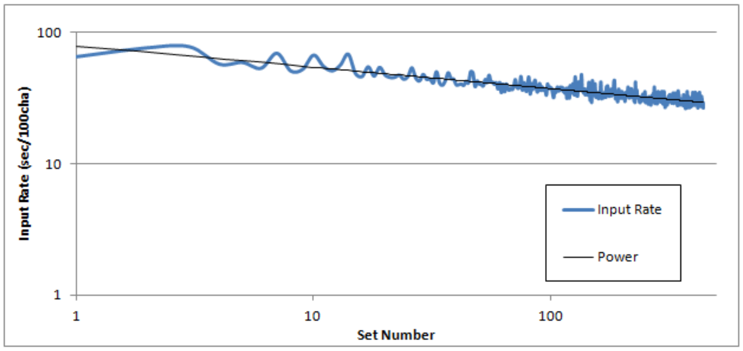
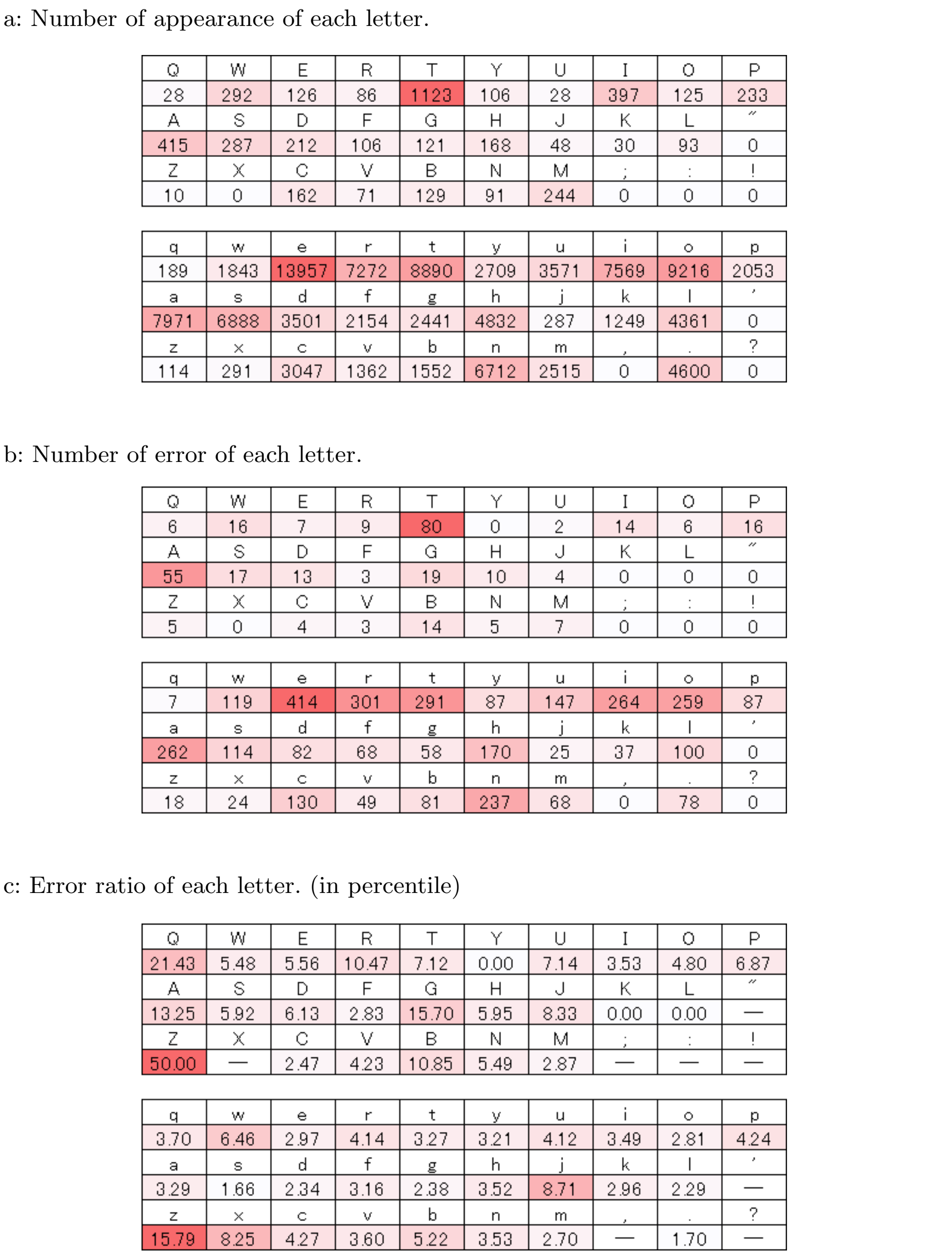


Fig. 10. Input rate (seconds per 100 letters) on the Meyboard.



a: Number of appearance of each letter.

Q	W	E	R	T	Y	U	I	O	P
28	292	126	86	1123	106	28	397	125	233
A	S	D	F	G	H	J	K	L	″
415	287	212	106	121	168	48	30	93	0
Z	X	C	V	B	N	M	;	:	!
10	0	162	71	129	91	244	0	0	0

q	w	e	r	t	y	u	i	o	p
189	1843	13957	7272	8890	2709	3571	7569	9216	2053
a	s	d	f	g	h	j	k	l	′
7971	6888	3501	2154	2441	4832	287	1249	4361	0
z	x	c	v	b	n	m	,	.	?
114	291	3047	1362	1552	6712	2515	0	4600	0

b: Number of error of each letter.

Q	W	E	R	T	Y	U	I	O	P
6	16	7	9	80	0	2	14	6	16
A	S	D	F	G	H	J	K	L	″
55	17	13	3	19	10	4	0	0	0
Z	X	C	V	B	N	M	;	:	!
5	0	4	3	14	5	7	0	0	0

q	w	e	r	t	y	u	i	o	p
7	119	414	301	291	87	147	264	259	87
a	s	d	f	g	h	j	k	l	′
262	114	82	68	58	170	25	37	100	0
z	x	c	v	b	n	m	,	.	?
18	24	130	49	81	237	68	0	78	0

c: Error ratio of each letter. (in percentile)

Q	W	E	R	T	Y	U	I	O	P
21.43	5.48	5.56	10.47	7.12	0.00	7.14	3.53	4.80	6.87
A	S	D	F	G	H	J	K	L	″
13.25	5.92	6.13	2.83	15.70	5.95	8.33	0.00	0.00	—
Z	X	C	V	B	N	M	;	:	!
50.00	—	2.47	4.23	10.85	5.49	2.87	—	—	—

q	w	e	r	t	y	u	i	o	p
3.70	6.46	2.97	4.14	3.27	3.21	4.12	3.49	2.81	4.24
a	s	d	f	g	h	j	k	l	′
3.29	1.66	2.34	3.16	2.38	3.52	8.71	2.96	2.29	—
z	x	c	v	b	n	m	,	.	?
15.79	8.25	4.27	3.60	5.22	3.53	2.70	—	1.70	—

Fig. 11. Results of appearance in the user study. Contrast represents size of the value.

2. If a letter in a phrase differs from the input, then the algorithm focuses on the next letter of the input. If this matches the letter from the phrase, then the input is identified as an omission. Otherwise, it is identified as a mistake.
3. When an error is detected, the algorithm observes the next input letter and then ignores every further input until the correct letter. This is to avoid falsely identifying inputs following an omission with multiple mistakes.

For example, if the text was "force" and the input was "frce", then the algorithm would detect the omission of the 'o'. As another example, if the text was "force" and the input was "firce", then the algorithm would detect the mistake concerning the 'o'.

As a result of this identification process, errors can be categorized into the following patterns:

Pattern 1: Mistype capital letters as small letters (e.g., mistype A as a.)
Pattern 2: Mistype flick as tap (e.g., mistype z as a).
Pattern 3: Mistype tap as flick (e.g., mistype G as B).
Pattern 4: Mistype as the other row (e.g., mistype w as s).
Pattern 5: Mistype as the other column (e.g., mistype p as o).
Pattern 6: Unnecessary input from index or middle finger (e.g., mistype R as T).
Pattern 7: Lack of input from index or middle finger (e.g., mistype b as v).
Pattern 8: Omission.
Pattern 9: Others.

Moreover, error patterns occurring with high rates (over 1.00%, and occurring more than twice) are presented in Table 1 and Table 2.

Most often, the participant made a mistakes by performing the wrong actions on the correct key required for the input. Mistakes between the keys (i.e., Patterns 4 and 5) also occurred, but not significantly often. The low error rate for Patterns 4 and 5 indicates that the calibration of the Meyboard works effectively. In particular, the low error rate for Pattern 4 (the highest was mistyping w as s, with an error rate of 3.31%) demonstrates the suitability of our assumption in designing the Meyboard such that the user can precisely locate two keys in each column without seeing them.

However, the results also show that mistakes involving typing keys from different rows still occur, especially for the bottom row, which requires flicks for the input (pattern 2). Pattern 2 shows a high error rate for letters using the left index or middle fingers. It appears that it may be necessary to reconsider the threshold for identifying flicks (moving the finger by more than 2.8 mm) for these fingers.

Mistakes also occur for inputs involving the use of both the index and middle fingers simultaneously, which exist in the TopCenter, MiddleCenter, and BottomCenter areas (Patterns 6 and 7). It appears that the participant struggled to switch between using one or two fingers correctly. It will be a subject of our future work to investigate whether the errors in patterns 6 and 7 decrease as the user adapts to the Meyboard.

Table 1. Error patterns and rates for capital letters.

Error	Correct letter	Error rate	Pattern
A	Z	50.00	2
q	Q	17.86	1
a	A	11.33	1
g	G	9.09	1
T	R	5.81	6
t	T	4.99	1
H	J	4.17	6
s	S	3.83	1
d	D	3.77	1
w	W	3.08	1
(omission)	P	2.58	8
G	H	2.38	9
G	B	2.33	2
V	B	2.33	7
H	N	2.20	2
n	N	2.20	1
B	G	1.65	3
F	G	1.65	7
quote	O	1.60	9
W	E	1.59	5
Space	H	1.19	9
O	I	1.01	5

Table 2. Error patterns and rates for small letters.

Error	Correct letter	Error rate	Pattern
a	z	9.65	2
s	x	6.19	2
(omission)	z	5.26	8
s	w	3.31	4
(omission)	j	3.14	8
u	j	2.79	4
z	q	2.65	3
d	c	2.49	2
h	j	2.09	6
quote	p	1.95	4
u	y	1.48	7
y	h	1.35	4
comma	n	1.31	7
n	m	1.27	6
v	b	1.22	7
g	b	1.16	2
c	b	1.10	7
d	e	1.09	4
(omission)	q	1.06	8
r	t	1.03	7
e	x	1.03	9
t	r	1.00	6
c	d	1.00	3

It is also clear that the mistyping of capital letters as small letters (Pattern 1) occurred with a high error rate. We examined the logs, and determined that either the participant failed to input the shift (by placing four fingers) or the order of the input was reversed. For example, the participant frequently mistyped A as a by placing their four fingers after tapping the a. We suspect this is because placing four fingers is more difficult than tapping or flicking using one or two fingers. However, we designed the Meyboard to perform calibration frequently by assigning the shift input to the calibration process. Therefore, there is a possibility that this mistake is a drawback of the Meyboard, which should be further investigated.

Omissions (Pattern 8) mostly occurred in letters that were input using the left little and medicine fingers, especially during continuous inputs by the same hand (e.g., "iza" as "ia" and "two" as "to"). It will be a subject of our future work to investigate the causes of omissions.

State of the art soft keyboards incorporate error-collection functions based on language models. The accuracy of the Meyboard may also be improved if by implementing such functions.

6 Conclusions

In this paper, we have presented the Meyboard. This is a soft keyboard that enables touch-typing, which is difficult to achieve using conventional soft keyboards.

The Meyboard facilitates touch-typing by omitting the bottom row of the QWERTY layout, and adjusting the positions and sizes of the remaining keys to fit the user's fingers. The user employs tapping and flicking actions using one or two fingers to input 30 characters using 20 keys. Adjustment of the positions and sizes of the keys on one side of the Meyboard occurs whenever the user places down four fingers from a single hand. This input action is also employed for inputting the shift, space, enter, and backspace, which are used frequently whilst typing phrases.

A user study found that a participant using the Meyboard was capable of touch-typing at a maximum input rate of 45.6 wpm and the mean input rate in the final week of the study were 40.7 wpm, which is promising in comparison with related work.

The logs of the input from the user study show that the key adjustments of the Meyboard have a positive effect in reducing errors resulting from typing keys from the wrong column. However, it is also seen that particular input actions in the Meyboard, such as the tap, place-and-hold, or flick with two or four fingers, appear to introduce new patterns of errors.

Our future work will involve determining the causes of these new errors and attempting to reduce their occurrences. Moreover, we plan to conduct a further user study involving more participants, in order to compare the performance of the Meyboard with existing soft keyboards.

References

1. Azenkot, S., Wobbrock, J.O., Prasain, S., Ladner, R.E.: Input finger detection for nonvisual touch screen text entry in Perkinput. In: Proceedings of Graphics Interface 2012. pp. 121–129. GI '12 (2012)
2. Coskun, T., Wiesner, C., Artinger, E., Benzina, A., Maier, P., Huber, M., Grill, C., Schmitt, P., Klinker, G.: Gestyboard 2.0: A gesture-based text entry concept for high performance ten-finger touch-typing and blind typing on touchscreens. In: Proceedings of the 2nd International Conference on Human Factors in Computing and Informatics. pp. 680–691. SouthCHI '13 (2013)
3. Findlater, L., Wobbrock, J.: Personalized input: improving ten-finger touchscreen typing through automatic adaptation. In: Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems. pp. 815–824. CHI '12 (2012)
4. Gentner, D.R.: Keystroke timing in transcription typing. In: Cooper, W.E. (ed.) Cognitive Aspects of Skilled Typewriting, pp. 95–120. Springer-Verlag (1983)
5. Go, K., Endo, Y.: CATKey: Customizable and adaptable touchscreen keyboard with bubble cursor-like visual feedback. In: Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction. pp. 493–496. INTERACT '07 (2007)
6. Go, K., Tsurumi, L.: Arranging touch screen software keyboard split-keys based on contact surface. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems. pp. 3805–3810. CHI EA '10 (2010)
7. Gunawardana, A., Paek, T., Meek, C.: Usability guided key-target resizing for soft keyboards. In: Proceedings of the 15th International Conference on Intelligent User Interfaces. pp. 111–118. IUI '10, ACM New York, NY, USA (2010)
8. Li, F.C.Y., Guy, R.T., Yatani, K., Truong, K.N.: The 1Line keyboard a QWERTY layout in a single line. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. pp. 461–470. UIST '11 (2011)
9. MacKenzie, I.S., Soukoreff, R.W.: Phrase sets for evaluating text entry techniques. In: CHI '03 Extended Abstracts on Human Factors in Computing Systems. pp. 754–755. CHI EA '03 (2003)
10. Newell, A.: Mechanisms of skill acquisition and the law of practice. pp. 1–55. Lawrence Erlbaum Associates (1981)
11. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: BrailleType: unleashing braille over touch screen mobile phones. In: Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part I. pp. 100–107. INTERACT '11 (2011)
12. Sax, C., Lau, H., Lawrence, E.: LiquidKeyboard: An ergonomic, adaptive QWERTY keyboard for touchscreens and surfaces. In: Proceedings of the Fifth International Conference on Digital Society. pp. 117–122. ICDS '11, XPS (2011)
13. Southern, C., Clawson, J., Frey, B., Abowd, G., Romero, M.: An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. In: Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services. pp. 317–326. MobileHCI '12 (2012)
14. TinqWill: C# Tips (Japanese), http://www.geocities.jp/tinqwill/cs.html