Electronic Preprint for Journal of Information Processing Vol.23 No.5

Regular Paper

Probabilistic Touchscreen Text Entry Method Incorporating Gaze Point Information

Toshiyuki Hagiya[1,a)] Tsuneo Kato[2,b)]

Received: December 8, 2014, Accepted: June 5, 2015

Abstract: We propose a novel probabilistic text entry method that takes into account the influence of the distance between a gaze point and a touch position in order to improve typing efficiency. The proposed method dynamically changes the size of the search space for predicting candidate words based on a model that estimates the magnitude of touch position errors according to the distance between the gaze point and the touch position. This makes it easier for users to type intended words even when they glance at different areas on the screen. The performance of the method was evaluated in terms of input accuracy in total error rate (TER) and of typing speed in words per minute (WPM). The results showed that the proposed text entry method successfully reduced the TER by 19.4% and increased WPM by 12.3% compared to the conventional method.

Keywords: Text entry, mobile devices, eye tracking, gaze-supported interaction, multi-modal interface

1. Introduction

In accordance with the recent popularity of smartphones and tablets, touchscreen keyboards have also become popular. While a touchscreen display provides the benefits of a large display area and a flexible software keyboard in terms of its orientation, language switching capability and the key layout, a number of users still prefer physical keyboards because they make fewer errors [1], [2]. Major causes of errors are the lack of tactility and the small keys [3].

One solution is to adjust the target areas according to the actual touch distributions. Many studies have proposed different keyboards based on the actual touch distributions as described in the next section [4], [5], [6], [7], [8]. Another solution is to show a number of candidate words guided by sequences of characters associated with the typed characters. This approach is effective for users who select candidate words frequently. Moreover, it might be useful for such users to be provided with a larger selection of candidate words. On the other hand, the drawback of this approach is that a long list of irrelevant words is confusing when typing accurately, and searching for target words from the list is time consuming. To show appropriate candidate words by taking touch precision into account can reduce the operational load on users. There are various factors contributing to the touch precision, such as a display size and user's posture. One of the important factors is the gaze point. Users often type text while glancing at different areas of the keyboard, such as during typing to check whether the typed text is correct or not, and while referring to other texts. The touch precision decreases under these circumstances because users cannot recognize the key positions accurately. We measured the touch position error from a target associated with the gaze point quantitatively, and we found the variance of the touch position error increases in accordance with the distance between the gaze point and the touch target [9].

In this paper, we propose a novel probabilistic text entry method incorporating a model that estimates the magnitude of the touch position error according to the distance between the gaze point and the touch position (hereinafter referred to as the "gaze-touch model") [9]. The size of the search space for candidate words changes dynamically in association with the touch position error estimated by the gaze-touch model. More specifically, the size of the search space for an input is minimized when a user types carefully with a steady gaze as shown in **Fig. 1** (a), whereas it increases when a user types carelessly while glancing at different areas as shown in Fig. 1 (b).

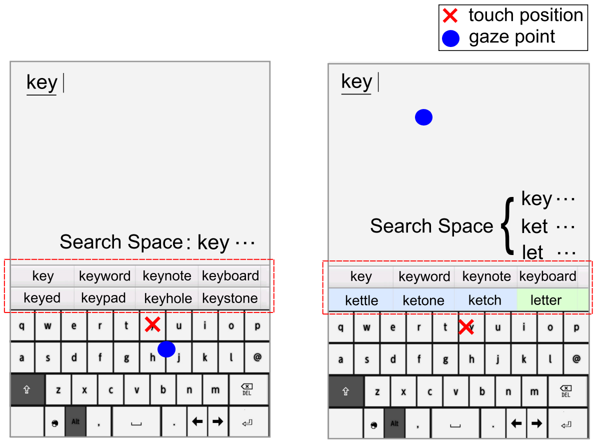
This paper is organized as follows. First, we present an overview of related work. Second, we describe the proposed technique in detail. Then, we describe an experiment for determination of the gaze-touch model. Next, we evaluate the performance of the proposed text entry method. Finally, following the discussion, we present our conclusions.

2. Related Work

2.1 Text Entry Method with Touch

Previous studies on reducing typing errors fall into two categories: language-model-based approaches and touch-model-based approaches. In the language-model-based approaches, the characters or words to be input next are predicted and displayed [10], [11]. In one study, the keys with high probabilities were displayed with bigger key sizes [12]. However, the performance of the language-model-based approach is heavily dependent on the dictionary. Unregistered words, such as new words,

1 KDDI R&D Laboratories, Inc., Fujimino, Saitama 356–8502, Japan
2 Doshisha University, Kyotanabe, Kyoto 610–0394, Japan
a) to-hagiya@kddilabs.jp
b) tsukato@mail.doshisha.ac.jp



(a) Accurate touch while gaze is in key area

(b) Ambiguous touch while gaze is in another area

Fig. 1 The image of the probabilistic text entry method with gaze-touch model while gazing at the key area and while glancing at other areas.

abbreviations, or colloquial expressions, are ignored. In contrast, in touch-model-based approaches, some studies proposed transforming the keyboard layout according to the actual touch distributions [13], [14]. In these studies, the center of each key was moved to follow the centroid of the actual touch distributions, and the boundary was set in the middle of the keys. However, that method did not consider key context, i.e., the previous key. The studies also reported that some users were confused by the dynamic changes in the key layout.

Furthermore, some studies proposed methods for transforming the detection areas based on both the language-model-based and touch-model-based approaches. The probability of a particular character to be input next was modeled by the product of two probabilities, viz. one from the language model and the other from the touch model [4], [5]. Goodman et al. [5] used character n-grams for the language model and the Gaussian mixture model (GMM) of stylus input positions in the touch model. Asrla et al. [4] proposed methods for transforming the detection areas based on the typing history in the language model, although the central region of the key remained fixed to suppress excessive transformation.

Practical issues in the real environment have also been tackled. Some studies analyzed the touch distributions for different hand postures, and proposed a keyboard that adapted to the users and the hand postures [8], [15]. Goel et al. [15] proposed switching various keyboard models by detecting the hand posture based on tap sizes and elapsed time between taps. Yin et al. [8] defined the hierarchical submodels depending on the user and the hand posture in order to adapt each model properly according to the amount of training data. Some studies analyzed the differences in touch distributions between sitting and walking, and they proposed adaptation techniques for these contexts [7].

2.2 Input with Gaze

Eye typing, which inputs text by using the focus of the gaze, has been researched for more than three decades [16], [17]. Although eye typing has the advantage that users do not have to use their hands, they experience eye fatigue, resulting in a decrease in performance [16].

Some methods for improving input usability combined touch and gaze information because the performance obtainable by eye tracking alone was not good enough. Zhai et al. [18] presented the MAGIC (Manual And Gaze Input Cascaded) pointing method that is a combination of mouse and gaze input for fast item selection. The idea is to move the cursor in the vicinity of the user's point of interest prior to moving the mouse. Then the cursor can be manually positioned using the mouse for more precise selections. Bieg et al. [19] analyzed the relationship between the positions of the eyes and the mouse pointer in a selection task and found that users automatically minimize the distance to probable targets. Stellmach et al. [20] proposed a method in which the nearest object from the gaze point was selected when a user touched some overlapping objects. Nagamatsu et al. [21] constructed an eye tracker with a mobile device and propose the system with which user could move the cursor by eye and select an object by simply touching anywhere. The system made it easier to operate a mobile device in one hand. As stated above, many studies combined touch and gaze information. However, to date, the relationship between the gaze point and the distribution of touch has not been studied.

3. Probabilistic Text Entry Method with Gaze-Touch Model

The text entry method that we propose here has two features. The first feature is that the gaze-touch model, which models the relationship between the variance of touch position error and the distance between the gaze point and touch position, is incorporated into the probabilistic text entry method. One of the other approaches using gaze information for touchscreen text entry is to use its information directly as the input feature vector along with the touch position when estimating input character by probabilistic methods. However, the drawback of this approach is that more data is needed for training or adapting to users due to the increase in size of the feature vectors. Therefore, we choose to construct the gaze-touch model independently.

The second feature is that the number of character sequences for searching candidate words changes dynamically according to the occurrence probability of a character that is controlled by the gaze-touch model.

3.1 Formulation of the Probabilistic Text Entry Method

First, we explain a conventional probabilistic text entry method. The occurrence probability of a character is given by the product of the probability of the touch model and that of the language model using Bayes' theorem as in

$$
\begin{aligned}
P(c_j|(x_t, y_t)) &= \frac{p((x_t, y_t)|c_j) \cdot p(c_j)}{p(x_t, y_t)} && (1)\\
&\propto p((x_t, y_t)|c_j) \cdot p(c_j) && (2)\\
&= p_T \cdot p_L && (3)
\end{aligned}
$$

where $P(c_j|(x_t, y_t))$ means the probability of a character c_j in the case of input of the coordinate (x_t, y_t), p_T is the probability of the touch model and p_L is that of the language model. The probability of touch model p_T can be represented by a single Gaussian distribution in the simplest case. In practice, the performance was improved by a Gaussian Mixture Model (GMM) because the touch distribution is different depending on hand postures, such as the typing hand. The generative model outputs the probability p_T using the following equation

$$p_T = \sum_{k=1}^{K} \omega_k N((x_t, y_t)|\mu_{jk}, \Sigma_{jk}) \tag{4}$$

$$\Sigma_{jk} = diag(\sigma_x^2, \sigma_y^2) \tag{5}$$

where N denotes a Gaussian distribution with a mean vector μ_{jk} and a diagonal covariance matrix Σ_{jk} and K is the number of mixtures. On the other hand, the language model, which formulates the probabilities of the subsequent characters, is implemented by a character 3-gram because the probability of c_j greatly depends on the history of character sequences. The probability p_L is shown as follows

$$p_L = p(c_j|h) \tag{6}$$

where h is the history of the character sequences. The probability of a sequence is given by the product of the probabilities of all the characters in the sequence.

3.1.1 User Adaptation

The touch model is adapted to an individual user based on the maximum likelihood linear regression (MLLR) algorithm that has been reported to be effective for key input [6]. In MLLR, the means of the normal distributions are transformed and re-estimated as

$$\mu'_k = \mathbf{A}\mu_k + b \tag{7}$$

where μ_k is the mean vector of Eq. (4), and μ'_k is the adapted mean vector. ($\mathbf{A}$, b) are obtained by maximum-likelihood estimation.

3.2 Formulation of the Gaze-touch Model

In the human visual system, the fovea, the part responsible for visual acuity, spans 1–2 degrees of the visual field [22], [23], and the spatial resolution decreases as the distance from the fovea increases. Touch position error (δ), which is the distance between the touch position and a target, can be expected to increase when users look at a point farther from the target. Consequently, we formulate the gaze-touch model such that it gives the variance of the touch position error (σ_δ^2) based on the distance (L) between two coordinates (x_g, y_g) and (x_t, y_t). Since it is known that the central visual field is a concentric circle with an angle of 20–30 degrees or an ellipse with a horizontal angle of 30 degrees and a vertical angle of 20 degrees [23], we propose the two gaze-touch models as shown in **Fig. 2**. One is the "concentric circular model" (model A) and the other is the "elliptic model" (model B). Let us formulate the gaze-touch models as shown in the following equations:

$$\sigma_{\delta_k}^2 = f(L) \tag{8}$$

$$L = \sqrt{\alpha^2(x_g - x_t)^2 + (y_g - y_t)^2} \tag{9}$$

$$\alpha = \begin{cases} 1 & (modelA) \\ \frac{\tan 20^\circ}{\tan 30^\circ} & (modelB) \end{cases} \tag{10}$$

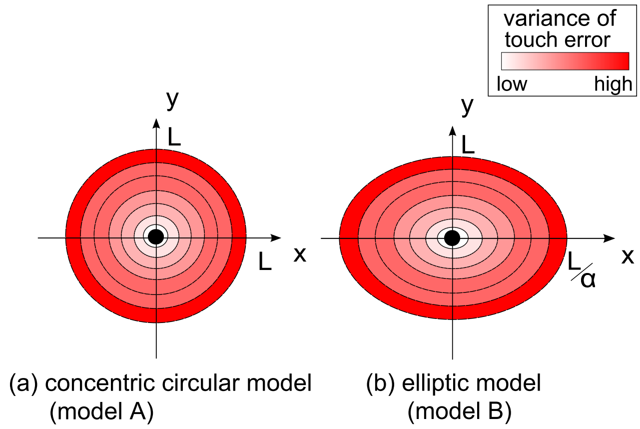


Fig. 2 Gaze-touch model.

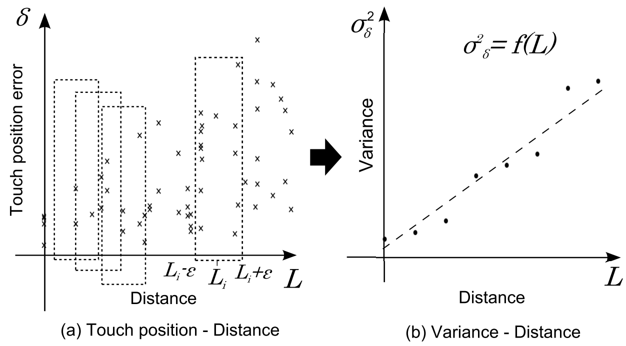


Fig. 3 Conversion from the touch position error to the variance.

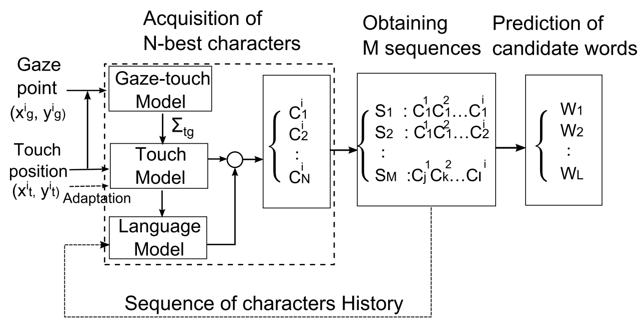


Fig. 4 Block diagram of the proposed method.

where f is a monotonically increasing function, k corresponds to either of x or y, and α is the scaling factor for axis lengths. **Figure 3** shows the operation for calculating $\sigma_{\delta_k}^2$ as the variance of δ_k in a moving window. The function $f(L)$ in Eq. (8) is determined by a regression.

3.3 Incorporating the Gaze-touch Model Into the Probabilistic Text Entry Method

First of all, we explain the abstract of the proposed method by using **Fig. 4**, which shows a block diagram of the approach. First, N-best characters are output based on the probabilistic text entry method incorporated into the gaze-touch model, which is the main contribution of this paper. Then, sequences of characters are obtained from the possible combinations of characters. Finally, the candidate words are predicted by the sequences.

Next, we explain each block in Fig. 4. The coordinates (x_g^i, y_g^i) and (x_t^i, y_t^i) are the gaze point and the touch position on the screen, respectively. The superscript i corresponds to the number in a character sequence. The gaze-touch model outputs an estimated variance of the touch position based on the inputs (x_g^i, y_g^i) and (x_t^i, y_t^i). The touch model is formulated by a GMM for each key. In our method, the variances of the gaze-touch model are used to take into account the influence of the gaze point. The effect of L can be taken into account if we simply replace the variance of the conventional model Σ_{jk} by Σ_{tg}. Specifically, the touch model incorporating the gaze-touch model is shown by the following equation

$$p_T = p((x_t, y_t)|c_j, L) = \sum_{k=1}^{K} \omega_k N((x_t, y_t)|\mu_{jk}, \Sigma_{tg}) \quad (11)$$

$$\Sigma_{tg} = diag(\sigma^2_{\delta x}, \sigma^2_{\delta y}) \quad (12)$$

where $\sigma^2_{\delta_x}$ and $\sigma^2_{\delta_y}$ are given by Eq. (8). The occurrence probability of every character is given by the product of the probabilities of the touch model and of the language model; in this case, character n-grams were used to create the model.

The characters of variable N-best are output according to their probability. The number N changes depending on how many characters have a probability within a range from the highest probability among all the characters as in

$$N = |\{c_j \,|P(c_j) > r_{th} \cdot P_{\max}\}| \quad (13)$$

where r_{th} is a threshold between 0 and 1. The larger the variances from the gaze-touch model are, the smaller are the differences between the probabilities of the most probable character and the probabilities of the others. The number of output characters, N, increases as a result. The probability of a sequence of characters is given by the product of the probabilities of the characters in the sequence.

Next, M-sequences of characters with high probabilities are obtained from the possible combinations of characters. The probabilities of sequences are given as the product of the probabilities of the characters in the sequence. The actual figures in M are determined by the preliminary experimental results. Finally, candidate words are predicted by left-hand match to the M-sequences based on the predicting function of OpenWnn.

4. Determination of Gaze-touch Model

In this section, we determine the function of the gaze-touch model, i.e., $f(L)$ in Eq. (8), and evaluate the goodness of the modeling in terms of the determination coefficients (R^2).

4.1 Data Collection

We recruited sixteen participants (ten males and six females) between the ages of 25 and 59 (average 35.1). Thirteen of them (eight males and five females) were right-handed. All of them were familiar with typing with a smartphone, although they did not have experience with gaze-based interfaces. The mobile device was the AQUOS PAD SHT21 running Android 4.0 with a 7.0 inch screen having a resolution of 1,280 × 800 pixels (216 ppi). The participants were instructed to touch a target (touch-target) while looking at another point (gaze-target). The coordinate of the gaze-target was used as the true value of the gaze point instead of a measured position obtained by an eye tracker. The

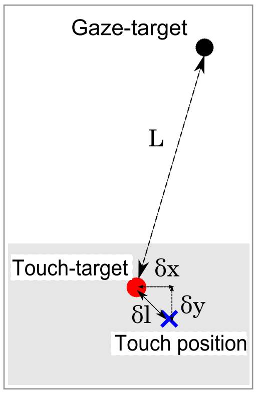


Fig. 5 Screenshot of the application for the determination of the gaze-touch model.

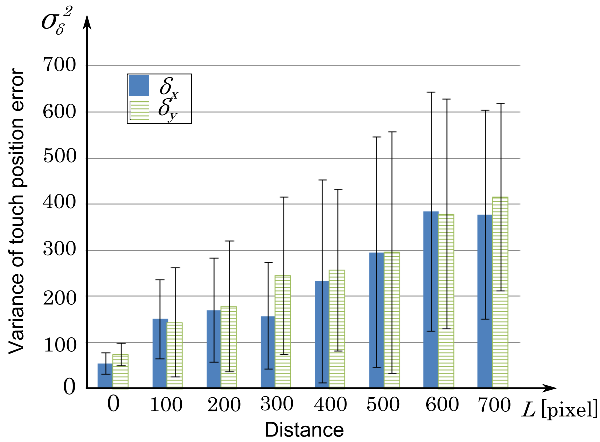


Fig. 6 Average of the variances of the "concentric circular model" (Model A); vertical bars show standard errors.

touch-target and gaze-target were depicted respectively as a red dot and a black dot with a radius of 15 pixels as shown in **Fig. 5**. Participants touched eight times per set and repeated ten sets of different gaze-targets. They operated the mobile device holding the device with both hands while sitting in a comfortable position on a chair. The touch-target changed its position in the area corresponding to the keyboard after every touch, while the gaze-target was fixed during a set. The mobile device recorded the coordinates of the touch position (x_t, y_t) in addition to those of both targets.

4.2 Determination of the Functions of the Gaze-touch Model

We compared the functions of the "concentric circular model" (Model A) to those of the "elliptic model" (Model B) for the variances of δ_x and δ_y. The variances were computed with a moving window. The window size and the window shift for calculating the variance were 200 pixels and 100 pixels, respectively. **Figure 6** shows the relation between L and the measured variance of each model. In general, as we had anticipated, the larger L was, the larger the variance was.

For modeling the relation between L and the variance of two models, three regressions, i.e., linear regression, exponential regression with a natural basis, and logistic regression, were examined. **Figure 7** shows determination coefficients, R^2, for each variance and regression. The R^2s of the regression functions for

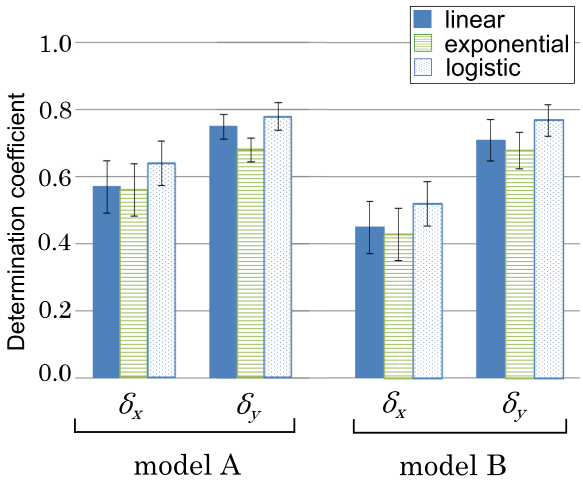


Fig. 7 Average determination coefficient of each model of regression analysis; vertical bars show standard deviations.

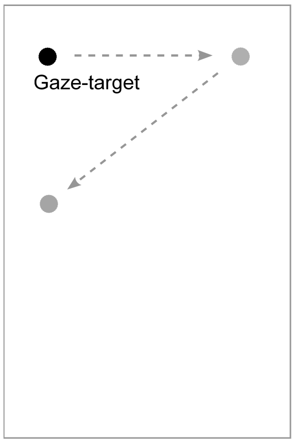


Fig. 8 Screenshot of the application for the gaze point calibration.

Table 1 The deviations of the gaze estimation immediately after the calibration; standard deviation in parentheses.

Participant	Horizontal accuracy [pixels]	Vertical accuracy [pixels]
1	75.4 (24.3)	89.8 (23.4)
2	100.0 (25.5)	119.0 (25.3)
3	36.6 (24.9)	75.9 (23.7)
4	34.4 (27.6)	77.3 (26.3)
5	64.5 (30.1)	86.4 (26.0)
Average	62.1 (26.6)	86.4 (25.7)

each model show very little difference. ANOVA did not show a significant difference ($F(6, 90) = 2.31, p \leq 0.05$). Accordingly, we adopted the linear regression of Model A, the simplest model. Hereinafter, the regression is defined as $\sigma^2_{\delta_x} = aL + b$ ($a = 0.46, b = 19.7$) and $\sigma^2_{\delta_y} = cL + d$ ($c = 0.46, d = 37.7$).

5. Evaluation of the Probabilistic Text Entry Method Incorporating the Gaze-touch Model

In this section, we evaluate the performance of the probabilistic text entry method incorporating the gaze-touch model. In our method, it is essential to estimate the gaze point in real time on a mobile device. Although some precise eye trackers such as Tobii Glasses [24] are being developed, most of them are limited in terms of their offline data recording capability. Accordingly, we constructed an eye tracker that can record in real time on a mobile device. First of all, we explain the construction of the eye tracker. After that, we describe in detail the performance evaluation of the proposed text entry method in comparison with two baseline methods: the conventional method, which has a fixed key detection area (B1), and the probabilistic method without the gaze-touch model (B2).

5.1 Estimating Gaze Point from a Built-in Camera on a Mobile Device

The participant's face was recorded using a built-in camera on the mobile device, and the facial and eye data were captured by Snapdragon SDK for Android in real time [25]. The obtained data were as follows: coordinates of eyes and mouth, face angle (pitch, roll and yaw), horizontal and vertical gaze angle value, and gaze point relative to the position of the camera. The gaze point on the screen is calculated with these pieces of information. The correspondence between the gaze point on the screen and these data was modeled by multiple linear regression before conducting each set of the typing experiment because the coefficients differ from the experimental environment. **Figure 8** shows a screenshot of the calibration procedure. Participants looked at a stationary gaze-target that moved after each trial. Participants repeated the trial 60 times. When all of the trials were completed, the coefficients of the multiple linear regression were calculated. The coefficients were used in each set of the experiment.

We recruited five participants (four males and one females) between the ages of 25 and 32 (average 28.6) for the evaluation of the accuracy of the gaze point estimation. None of the participants had previous experience with gaze-based interfaces. None of them wore glasses because if glasses are worn, the accuracy of the gaze point estimation deteriorates. The evaluation was done immediately after the calibration. The participants were instructed to keep looking at a series of gaze-targets which were displayed one after another randomly. They operated the mobile device holding the device with both hands while sitting in a comfortable position on a chair. Each gaze-target was depicted as a black dot with a radius of 15 pixels as shown in Fig. 8. The gaze-target was displayed for 5 seconds in each trial, and the trial was repeated 30 times in succession for different gaze-targets. The mobile device recorded the estimated gaze point (x_g, y_g) at intervals of 0.10 seconds. After the experiment, the horizontal and the vertical deviation were calculated using the central 2.5-second data. **Table 1** shows the accuracy of the gaze point estimation immediately after calibration. The horizontal and vertical deviations were 62.1 pixels and 86.4 pixels, respectively. Assuming the screen was viewed from a distance of 40 cm, the horizontal and vertical deviations were presumed to be around 1.1° and 1.5°, respectively. However, the performance is considered to decline with the passage of time because the eye tracking technique does not consider changes of the user's posture. Therefore, calibration is necessary for an experiment periodically. Although the performance was not outstanding even immediately after the calibration compared with the performance results obtained in the latest research [26], the eye tracking technique was used thereafter.

Fig. 9 Screenshot of the application for the evaluation experiment.

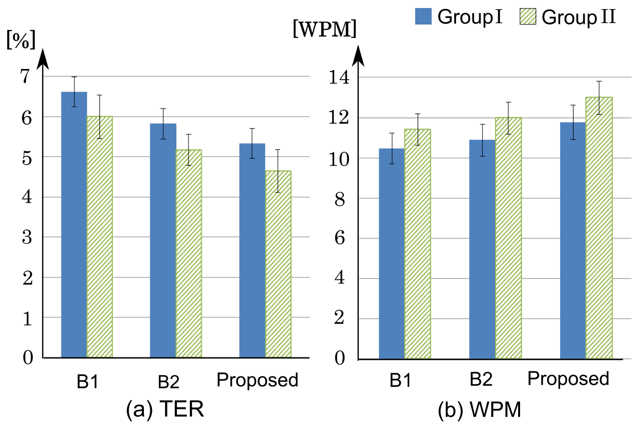


Fig. 10 Comparison of total error rate (TER) and words per minute (WPM); vertical bars show standard errors.

5.2 Data Collection and Evaluation of the Proposed Text Entry Method

We recruited eighteen participants (thirteen males and five females) between the ages of 23 and 40 (average 31.0). Fourteen of them (ten males and four females) were right-handed. None of them wore glasses. All of them were familiar with typing with a smartphone, although none of them had previous experience with gaze-based interfaces. The device was the same as that used in the previous experiment. The device recorded the gaze point (x_g, y_g) and the touch position (x_t, y_t), and calculated the touch error variance based on the gaze-touch model. The gaze point was measured in real time using the eye tracking technique described in the previous subsection. The participants typed using three text entry methods: the proposed method, the conventional method, which has a fixed key detection area (B1), and the probabilistic method without the gaze-touch model (B2). The keyboards all had the same QWERTY layout with a fixed key display area. The participants typed six sets of 20 phrases chosen from MacKenzie's phrase set [27] using each method with the typing application as shown in **Fig. 9**. They operated the mobile device holding the device with both hands while sitting in a comfortable position on a chair. The calibrations of the gaze point were carried out for every set before the input trial. The participants were permitted to correct their inputs and to select candidate words to input the exact-match phrases. The methods and the set of phrases were presented in random order.

For the two probabilistic methods, the character 3-gram of the language model was trained by maximum likelihood estimation (MLE) on the Brown corpus which consists of American English from a variety of sources [28]. The touch model was also trained by MLE with touch data for 840 phrases typed by participants (five males and two females) between the ages of 23 and 36 (average 29.2). They did not participate in the evaluation experiment. The number of mixtures of the GMM was set at 2. Data for the MLLR adaptation were the correctly typed data of the first phrase of each set. We evaluated the remaining 19 phrases except for the first phrase of each set. From the preliminary experimental results, the threshold r_{th} for deciding N of Eq. (13) was set at 0.5 and the maximum number of sequences of characters for searching candidate words, M, was set at 6. The candidate words were searched by left-hand match using OpenWnn [29].

The proposed method is considered to be of benefit to users who select candidate words frequently. The TER and WPM were tallied for two groups: all participants (Group I), and the upper half of participants in terms of the ratio of selecting candidate words (RSC) with text entry method B1 (Group II). The RSC was defined as the ratio of the number of candidate words selected to that of all words in the phrases.

5.3 Evaluation of the Proposed Text Entry Method

Figure 10 (a) and (b) show the average TER and WPM for the three methods. Looking at the results for Group I, the TER and WPM were 6.62% and 10.4 for the conventional method (B1), 5.83% and 10.9 for the probabilistic method (B2), and 5.33% and 11.8 for the proposed method. In other words, the proposed method reduced the TER by 19.4% and 8.4%, and increased WPM by 12.3% and 8.2% compared to the other two methods. The major factor influencing the improvement was that the typing counts and editorial operations were reduced owing to more hits on candidate words. Actually, the RSC of the proposed method increased compared to the others. The RSCs were 27.8% for B1, 32.0% for B2, and 37.9% for the proposed method. ANOVA revealed statistically significant differences among the three methods in TER ($F(2,34) = 3.34$, $p \leq 0.05$) and WPM ($F(2, 34) = 3.56$, $p \leq 0.05$). However, a post-hoc Tukey's test did not show any significant difference between B2 and the proposed method.

On the other hand, focusing on Group II in Fig. 10 (a) and (b), the proposed method reduced the TER by 22.5% and 11.4%, and increased WPM by 13.9% and 9.6% compared to the other methods. ANOVA showed statistically significant differences among the three methods in TER ($F(2, 16) = 3.92$, $p \leq 0.05$) and WPM ($F(2, 16) = 4.04$, $p \leq 0.05$). A post-hoc Tukey's test also showed significant differences between the proposed method and the other two. These results suggest that the proposed method benefited those who used the list of candidate words frequently.

6. Discussion and Future Work

A significant difference was observed in TER and WPM between the proposed method and method B2 for participants with a high RSC. It is possible to automatically categorize those users into a group based on RSC. The proposed method is apparently of benefit to users in this category, which implies that the method must be useful in practice. The effectiveness of the proposed method is particularly significant for long words as it reduced TER by 13.2%, and increased WPM by 14.0% compared to the probabilistic method for words with five or more letters. This is probably because the longer words are, the greater the time required for correction if the intended words are not displayed. The effectiveness of the proposed method is directly attributable to the difference of the word candidates due to the introduction of the gaze-touch model. However, the difference of the word candidates has various factors other than the gaze-touch model, such as r_{th} which is the threshold for deciding N of Eq. (13) and M which is the maximum number of sequences of characters for searching candidate words. It might be more effective to adapt the values to the users, though we adopted the likely values based on the preliminary experiment in this study. For example, to set the threshold r_{th} smaller might be effective for a user who often mistypes and selects candidate words.

In addition, the gaze-touch model would be improved if the performance of the eye tracking could be improved. The variance from the gaze touch model changes in proportion to the distance between the gaze point and the touch position. This means the variance is proportional to the gaze point estimation error. As a result, candidate words may change depending on the touch position and the gaze point estimation error. In this way, the accuracy of the gaze estimation directly affects usability. In this study, the eye tracking technique was very simple, though there are various factors contributing to the touch precision, such as the movement and the variance of the gaze point and the influence by blink. If the performance of the eye tracker improves, we can analyze the relationship. In fact, it is known that users tend to look at targets before clicking the mouse [30]. It is considered that taking into account the gaze point immediately before the touch operation would make the model more accurate because a similar phenomenon might also happen in the touch operation. Therefore, it is necessary to investigate the relationship between touch and gaze point more precisely.

On the other hand, the experiments were conducted only in the sitting position even though the distributions of touch and gaze point become large when the user is moving. In addition, users adopt a variety of hand postures when using mobile devices in actual practice. However, switching hand postures was not considered in our experiment although the touch model was adapted to users. A possible solution to this issue is proposed in Ref. [8]. They proposed detecting hand postures by touch distributions and changing models accordingly. By incorporating their methods and adopting a number of touch models for the hand postures, the proposed method could be further improved.

7. Conclusion

In this paper, we proposed a probabilistic text input method that took gaze point into consideration. First, the gaze-touch model, which models the relationship between the variance in the touch position error and the distance between the gaze point and the touch position, was formulated. Next, the model was incorporated into the Gaussian mixture distributions of the touch model of the probabilistic method. Experimental results showed that the proposed method reduced the total error rate by 19.4% and increased typing speed by 12.3% compared to the conventional method. As a future area of research, we will investigate in detail the relationship between touch and gaze point.

References

[1] Sears, A., Revis, D. and Swatski, J., et al.: Investigating touchscreen typing: The effect of keyboard size on typing speed, *Behaviour &Information Technology*, Vol.12, No.1, pp.17–22 (1993).

[2] Lee, S. and Zhai, S.: The performance of touch screen soft buttons, *Proc. CHI 2009*, pp.309–318, ACM (2009).

[3] Hoggan, E., Brewster, S.A. and Johnston, J.: Investigating the effectiveness of tactile feedback for mobile touchscreens, *Proc. CHI 2008*, pp.1573–1582, ACM (2008).

[4] Asela, G., Paek, T. and Christopher, M.: Usability Guided Key-Target Resizing for Soft Keyboards, *Proc. IUI 2010*, pp.111–118, ACM (2010).

[5] Goodman, J., Venolia, G., Steury, K. and Parker, C.: Language Modeling for Soft Keyboards, *Proc. IUI 2002*, pp.194–195, ACM (2002).

[6] Hagiya, T. and Kato, T.: Probabilistic Keyboard Adaptable to User and Operating Style Based on Syllable HMMs, *Proc. ICPR 2012*, pp.65–68, IEEE (2012).

[7] Goel, M., Findlater, L. and Wobbrock, J.O.: WalkType: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry, *Proc. CHI 2012*, pp.2687–2696, ACM (2012).

[8] Yin, Y., Ouyang, T.Y., Partridge, K. and Zhai, S.: Making Touchscreen Keyboards Adaptive to Keys, Hand Postures, and Individuals - A Hierarchical Spatial Backoff Model Approach, *Proc. CHI 2013*, pp.2775–2784, ACM (2013).

[9] Hagiya, T. and Kato, T.: Probabilistic Touchscreen Keyboard Incorporating Gaze Point Information, *Proc. MobileHCI 2014*, pp.329–333, ACM (2014).

[10] Tantug., A.C.: A Probabilistic Mobile Text Entry System for Agglutinative Languages, *IEEE Transactions on Consumer Electronics*, Vol.56, No.2, pp.1018–1024, ACM (2010).

[11] Nils, K. and Michael, R.: Word n-grams for cluster keyboards, *TextEntry'03 Proc. 2003 EACL Workshop on Language Modeling for Text Entry Methods*, pp.51–58 (2001).

[12] Khaldoun, A, F., Mustapha, M. and Nadine, V.: BigKey: A Virtual Keyboard for Mobile Devices, *Proc. 13th International Conference, HCI International 2009, Part III*, pp.3–10 (2009).

[13] Himberg, J., Hakkila, J. and Kangas, P., et al.: On-line Personalization of a Touch Screen Based Keyboard, *Proc. IUI 2003*, pp.77–84, ACM (2003).

[14] Leah, F. and Jacob, W.: Personalized input: Improving ten-finger touchscreen typing through automatic adaptation, *Proc. CHI 2012*, pp.815–824, ACM (2012).

[15] Goel, M., Jansen, A., Patel, S.N. and Wobbrock, J.O.: ContextType: Using hand posture information to improve mobile touch screen text entry, *Proc. CHI 2013*, pp.2795–2798, ACM (2013).

[16] Kari, J.R. and Saila, O.: Word n-grams for cluster keyboards, *An Exploratory Study of Eye Typing Fundamentals: Dwell Time, Text Entry Rate, Errors, and Workload*, pp.3001–3010 (2012).

[17] Majaranta, P. and Raiha, K.J.: Twenty Years of Eye Typing: Systems and Design Issues, *Proc. ETRA 2002*, pp.15–22 (2002).

[18] Zhai, S., Morimoto, C. and Ihde, S.: Manual and gaze input cascaded (MAGIC) pointing, *Proc. CHI 1999*, pp.246–253, ACM (1999).

[19] Bieg, H.J., Reiterer, H. and Bulthoff, H.H.: Eye and pointer coordination in search and selection tasks, *Proc. ETRA 2010*, pp.89–92, ACM (2010).

[20] Stellmach, S. and Dachselt, R.: Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets, *Proc. CHI 2013*, pp.285–294, ACM (2013).

[21] Nagamatsu, T., Yamamoto, M. and Sato, H.: MobiGaze: Development of a Gaze Interface for Handheld Mobile Devices, *Proc. CHI EA 2010*, pp.3349–3354 (2010).

[22] Heijl, A., Lindgren, G. and Olsson, J.: Normal Variability of Static Perimetric Threshold Values Across the Central Visual Field, *Arch Ophthalmol*, Vol.105, No.11, pp.1544–1549 (1987).

[23] Walker, H.K., Hall, W.D. and Hurst, J.W.: *Clinical Methods: The History, Physical, and Laboratory Examinations*, Boston, Butterworths (1990).

[24] Tobii Technology: available from ⟨http://www.tobii.com/⟩.

[25] Qualcomm: available from ⟨https://developer.qualcomm.com/⟩.

[26] Zhang, Z. and Cai, Q.: Improving cross-ratio-based eye tracking techniques by leveraging the binocular fixation constraint, *Proc. ETRA 2014*, pp.267–270 (2014).

[27] MacKenzie, I.S. and Soukoreff, R.W.: Phrase sets for evaluating text entry techniques, *Proc. CHI 2003*, pp.754–755, ACM (2003).

[28] Kucera, H. and Francis, W.N.: *Pattern Recognition And Machine Learning*, Brown University Press (1967).
[29] OMRON SOFTWARE Co., Ltd: available from ⟨http://www.omronsoft.com/⟩.
[30] Kumar, M., Paepcke, A. and Winograd, T.: EyePoint: Practical Pointing and Selection Using Gaze and Keyboard, *Proc. CHI 2007*, pp.421–430, ACM (2007).

Toshiyuki Hagiya received his B.E. and M.E. degrees from Kyoto University in 2008 and 2010, respectively. He is currently with KDDI R&D Laboratories, Inc. He has been engaged in research and development of spoken dialogue systems and interactive user interface. He is a member of IPSJ, ACM.

Tsuneo Kato received his B.E., M.E. and Ph.D. degrees from The University of Tokyo in 1994, 1996, and 2011, respectively. He joined Kokusai Denshi Denwa Co. Ltd. in 1996. He is currently an associate professor in the department of Intelligent Information Engineering, Doshisha University. He has been engaged in research and development of automatic speech recognition, spoken dialogue systems, and interactive user interface. He received IPSJ Kiyasu Special Industrial Achievement Award in 2011. He is a member of IPSJ, ASJ, IEICE and IEEE.