Improving Flick Input Accuracy via Personalized Region Classification

Tomoya Kikuchi
Department of Information and Communications Engineering
Kogakuin University
Tokyo, Japan
J022112@g.kogakuin.jp

Takeshi Kamiyama
School of Information and Data Sciences, Nagasaki University
Nagasaki, Japan
kami@nagasaki-u.ac.jp

Masato Oguchi
Department of Information Sciences
Ochanomizu University
Tokyo, Japan
oguchi@is.ocha.ac.jp

Saneyasu Yamaguchi
Department of Information and Communications Engineering
Kogakuin University
Tokyo, Japan
sane@cc.kogakuin.jp

Abstract—**Flick input is widely used for text entry on smartphones, especially for Japanese characters; however, current systems assume fixed keyboard region boundaries, which may not fully reflect actual user behavior. In this paper, through analysis of practical flick-input coordinate data, we show that input errors occur mainly in the specified regions in the first consonant-selection step, and that the optimal boundaries of consonant regions depend on users. Based on these observations, we propose an individualized character-inference method that constructs personalized Support Vector Machine (SVM) models to adaptively correct these boundaries. The method performs consonant inference using two-class SVMs to distinguish neighboring regions, followed by vowel inference using personalized five-class SVM models. We evaluated the proposed method using flick-input data from four subjects. Experimental results demonstrate that the average incorrect-inference rate decreased from 2.12% to 0.38%, i.e., reduced to about one-fifth (≈ 82% reduction). These findings indicate that incorporating individual characteristics is effective for improving character inference in flick-input.**

Keywords—Flick Input, SVM, Support Vector Machine, Android, Smartphones, Touch screens, User models

I. Introduction

Flick input is a commonly used text entry method on smartphones, especially widely used for entering Japanese characters. As with other input methods, the character the user intends to enter sometimes differs from the character recognized by the system, which can degrade the user experience.

In flick input, the region allocated to each character is displayed in a systematic manner. We hypothesize that these fixed boundaries are not necessarily optimal and, crucially, that the optimal regions are user-dependent. We assume that, by considering these individual differences, the user's intended input can be inferred more accurately.

In this paper, we first analyze practical flick-input coordinate data to confirm that errors are primarily due to consonant-region boundary issues. Based on this finding, we then propose an individualized character-inference method utilizing personalized Support Vector Machine (SVM) models to adjust these boundaries. We demonstrate the effectiveness of our proposed method through performance evaluation.

The situations where the character the user intended to input differs from the character recognized by the system are not referred to as "incorrect input by the user" but we refer to them as "incorrect inference by the system."

II. RELATED WORK

A. Flick Input

Flick input is a widely used method for text input on smartphones, particularly for Japanese characters. It efficiently leverages the consonant-vowel structure of Japanese, allowing all 50 characters to be entered using a limited number of keys [1]. Users swipe within designated regions on the screen to specify the character they intend to input. The operation consists of two steps. the *first step* and the *second step*, as illustrated in Fig. 1. The screen displays 12 regions, and the consonant of the character to be entered is determined by the region where the user begins the swipe-this is the first step. Consonants are assigned in Japanese order (*no-consonants, k, s, t, ...*), arranged from the top left, moving rightward and then downward. We define these as Region 0 to Region 11, following this order. Next, the vowel is determined by the direction of the swipe-this is the second step. The vowels *a, i, u, e,* and *o*, correspond to no swipe, left swipe, up swipe, right swipe, and down swipe, respectively.

In the example in Fig. 1, the swipe starts from Region 1, which corresponds to the consonant *k*, and moves upward, which corresponds to the vowel *u*. Then the system recognizes the input as "*ku*."

B. Improving Input Accuracy

Shida et al. [2] noted that, in smartphone touch-based text input, device size limitations and insufficient visual feedback on finger movements can cause input errors. They also found that errors usually occur near the intended key, and it is rare for the system to detect a key more than two regions away. Based on this, they proposed a probabilistic correction method using a 3×3 mask, where the probability of the detected key being the intended one is 50%, and each surrounding key is 6.25%. Corrections are made using estimates from continuous flick operations and contextual information, following a Bayesian filter approach. However, their method does not personalize spatial decision boundaries based on individual users' touch-coordinate distributions. In contrast, our method learns per-user

Identify applicable funding agency here. If none, delete this text box.

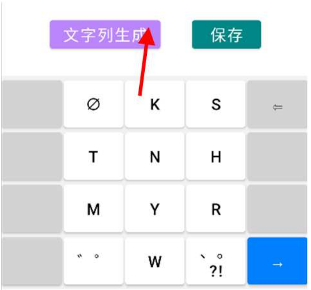

Fig. 1. Flick input

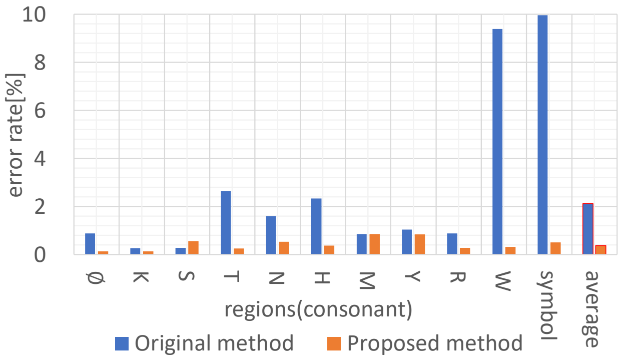

Fig. 2. Error date (regions)

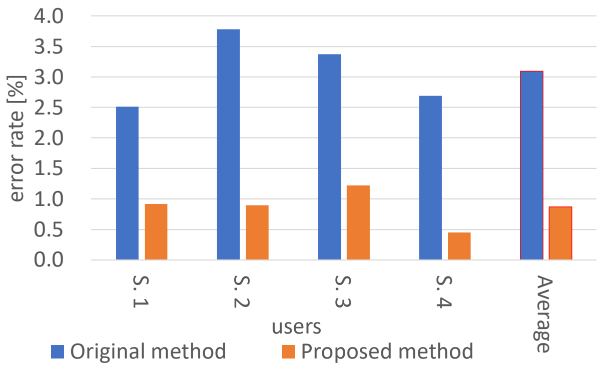

Fig. 3. Error date (users)

decision boundaries from past flick coordinates using SVM, enabling user-specific inference.

Gboard [3] provides correction features that suggest alternative words when user input may contain errors; these suggestions are drawn from dictionaries. However, its flick input mechanisms rely primarily on linguistic context (e.g., dictionary information) and do not directly adapt spatial region boundaries using raw swipe coordinates. Our approach complements such lexical correction by directly modeling users' coordinate deviations and adjusting region-level classification accordingly.

Sivek et al. [4] studied touchscreen keyboards for smartphones and analyzed practical touch coordinates. They found that most users rarely touch the physical center of a key on the screen. Based on their findings, they presented a personalized Gaussian spatial model that adapts key-center offsets to individual users, demonstrating modest gains in typing metrics on Gboard. Their focus, however, remains at the level of probabilistic spatial scoring of QWERTY taps rather than supervised region-level reclassification for flick gestures. Unlike that work, our method uses supervised SVM classifiers to directly decide between neighboring consonant regions for flick input, specifically targeting the two-step (consonant then vowel) flick process and its boundary-related errors.

III. Analyzing Practical User Flick Input Coordinates

In this section, we present the practical flick input coordinates of users and discuss both general tendencies across users and individual characteristics.

We asked four subjects to input 40 characters using flick input. Hereafter, we refer to them as *Subjects 1–4*. The character set consisted of Japanese hiragana characters and symbols. Characters were input using our flick input application that we developed for this study. As shown in Fig. 1, this application displays twelve regions corresponding to consonants, similar to Gboard. In our experimental device Pixel 3a, the screen resolution is 2220 × 1080 pixels, and each region measures 187.25 pixels in height and 214 pixels in width. The threshold for distinguishing between swipe and tap input is set at 44 pixels. These parameters are identical to those used in Gboard on the Pixel 3a.

Subject 1 placed the smartphone on a desk and input characters with their index finger. *Subject 2* held the smartphone in their hand and input with their right thumb. *Subject 3* held the smartphone with both hands and input with their right thumb. *Subject 4* held the smartphone with both hands and input with both thumbs.

The data labeled with "Original Method" in Fig. 2 show the ratios of inaccurate inference of input characters by the flick input system of each consonant in the first step. Note that the error rates in the second step were 0 for all vowels. Those in Fig. 3 depict the error rates of each user. “S. 1” to “S. 4” stand for “*Subject 1*” to “*Subject 4*.”

Figs. 4 to 7 show the touch coordinates on the screen when each subject entered the characters in the *a* row. For *a* row input, users must not perform a long swipe (i.e., the starting and ending touch coordinates should be close). Thus most flick trajectories appear nearly as single points. Each line, which looks like a dot, represents one input sample.

First, we examine general trends across users. From the figures, it can be observed that errors occur in the first step (consonant inference). Furthermore, the probability of incorrect inference depends on the regions. Specifically, the error rate is higher in region 10 for *w* and region 11 for symbols, approximately 10%.

Second, we discuss individual user trends. *Subject 2* tended to have more incorrect inferences near the boundary between the top regions (regions 0, 1, and 2) and second top regions (regions 3, 4, and 5), especially between regions 2 and 5. These results suggest that while some error locations are common across users, others differ. Therefore, learning input coordinates and

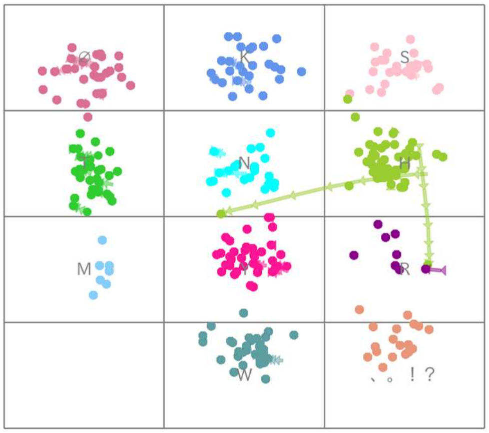

Fig. 4. Input coordinates for *a* column (*Subject 1*)

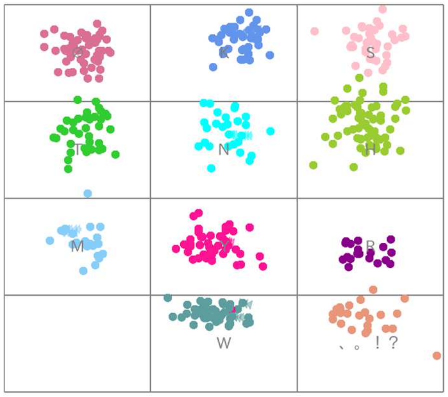

Fig. 5. Input coordinates for *a* column (*Subject 2*)

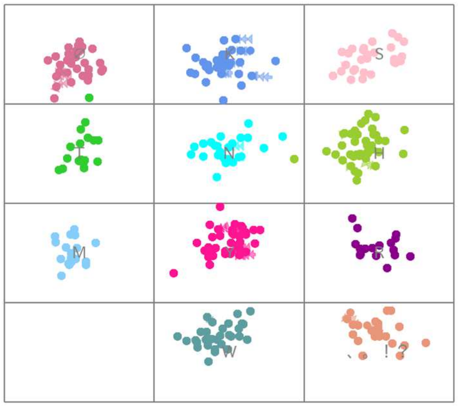

Fig. 6. Input coordinates for *a* column (*Subject 3*)

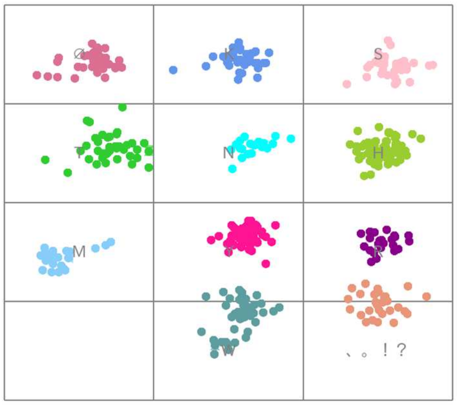

Fig. 7. Input coordinates for *a* column (*Subject 4*)

adjusting boundary taking individual differences into account are expected to be effective.

IV. Proposed Method

In this section, we propose to adjust the boundaries between regions based on each user's input coordinates collected in the past. Our method consists of two stages. two-class inference for consonant regions in the first step and five-class inference for vowels in the second step. We use SVM to determine the boundaries that distinguish regions. We define the coordinate of the start of the swipe as (x0, y0) and that of the end of the swipe as (x1, y1). For the first step, this method creates a model for two class classification for all pairs of neighboring regions. The explanatory variable is (x0, y0). The response variable is the region that the user intends to input. For the second step, the method creates a model for five-class classification. Model is constructed for each region because we assume that the difficulty of swiping depends on regions. The explanatory variable is (x0, y0, x1-x0, y1-y0). The response variable is the vowel that the user intends to input.

In the inference phase, the method infers the character that a user intends to input from a set of coordinates (x0, y0, x1-x0, y1-y0). For the first step, the method chooses the two nearest regions from the start point (x0, y0) as the candidates. Namely, the first region is the region that contains (x0, y0). The other is the nearest region other than the first candidate. The method then performs two-class classification between the two candidates with SVM and determines the consonant. For the second step, the method performs five-class classification with SVM based on (x0, y0, x1−x0, y1−y0) and determines the vowel.

V. Evaluation

In this section, we evaluate the performance of the proposed method. For the performance evaluation, we use the input data from the four participants described in Section 3. *Subject 1*'s dataset contains 42 sets of inputs. 34 sets and 8 sets were used for training and testing, respectively. *Subject 2*'s dataset contains 551 sets; 501 and 50 were used for training and testing, respectively. In the *Subject 3*'s dataset, 32 and 8 sets are used, and in the *Subject 4*'s dataset, 200 and 50 sets are used for training and testing, respectively. We note that the data size differs substantially among subjects, which may influence the per-subject performance. In the experiments, we target only inputting hiragana and do not consider kanji conversion.

The data labeled with "Proposed Method" in Fig. 2 and 3 depict the ratios of incorrect inference by regions and by users, respectively. The ratios of incorrect inference in all user and all regions with the original and proposed methods are 3.09% and 0.87%, respectively. These results show that the proposed method reduces the ratio of incorrect inference largely. The results in Fig. 2 demonstrate that the proposed method increases the accuracy especially in regions with high incorrect ratios such

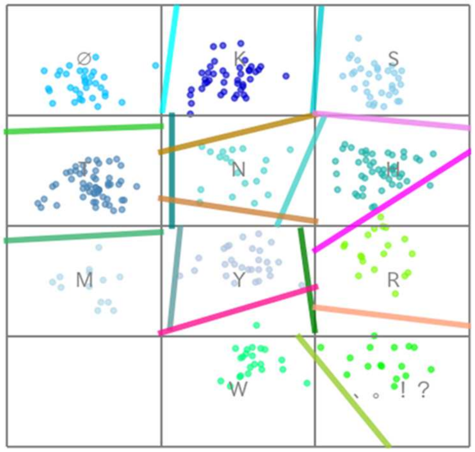

Fig 8. Adjusted boundary (*Subject 1*)

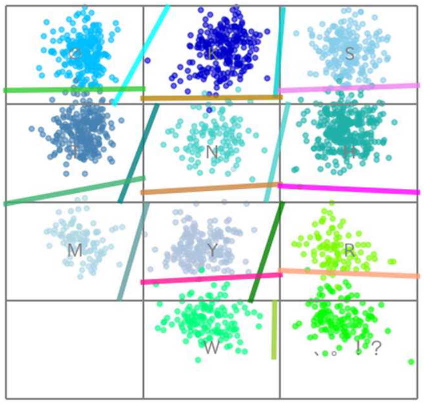

Fig 9. Adjusted boundary (*Subject 2*)

as regions 10 and 11. For reference, the incorrect inference ratios of each region for each subject are presented in the Appendix.

Figs. 8 - 11 show the adjusted boundary of the proposed method. For example, it can be observed that the boundary between the region 8 and 11 is moved up in all the subjects and this causes increase in the accuracy.

VI. Discussion

Our results indicate that the input characteristics depends on the individual user, but it may depend also on the device and on the how the device is held. For example, if the screen size becomes larger, swiping leftward or upward from Region 0, the Top-left region, using the right thumb will become more difficult and the input coordinates will be unstable. Therefore, we expect that the model should be constructed for each device and each holding way. Evaluating different devices and holding styles, and building models specific to each leave as future work.

In the cases of *Subjects 2* and *4*, where the amount of training data was large, a lower error rate was achieved. In contrast, in the case of *Subject 3*, where the amount of training data was smaller, the error rate was slightly higher. Therefore, we expect that collecting additional data for *Subject 3* would also lead to a lower error rate.

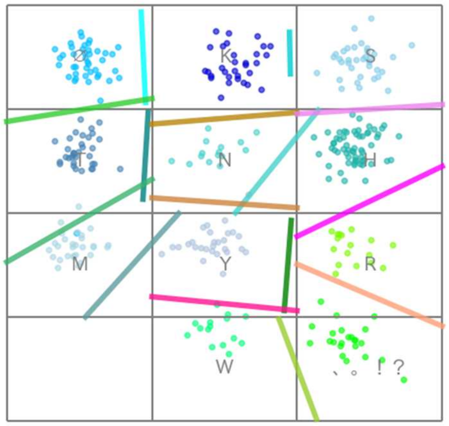

Fig 10. Adjusted boundary (*Subject 3*)

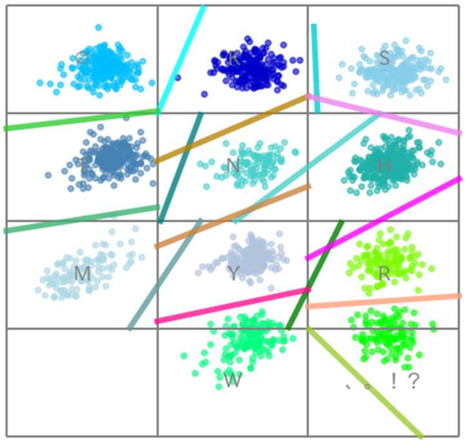

Fig 11. Adjusted boundary (*Subject 4*)

In this work, subjects were asked to input random character sequences, resulting in relatively low input speed. If users were asked to enter sentences they already know, the input speed would be higher, the deviation of swipe positions would increase. Consequently, the error rate would likely increase. In such cases, the effectiveness of the proposed method would also be expected to increase. Additionally, considering the dependencies on preceding character and on succeeding character for inference could further reduce the error rate.

VII. Conclusion

In this paper, we focused on incorrect inference of input character in smartphone flick input and analyzed practical swipe coordinate data from users. The analysis revealed that incorrect inference occurred mainly in the first step (consonant selection). Specifically, in the cases of our experiments, all incorrect inference occurred in this first step. Furthermore, we found that the regional boundaries defined by the system did not completely match with the effective boundaries reflected in users' touch coordinates. To address this issue, we proposed a method that uses an SVM to infer the character intended by the user based on touch coordinates. The proposed method builds a prediction model for each user, thereby taking user-specific input characteristics into consideration. We evaluated the estimation accuracy of the proposed method using practical flick input coordinates from users. As a result, the average error rate from 3.09% to 0.87%. These results demonstrate the effectiveness of personalized boundary estimation. Our method requires only past flick coordinates and no language model or dictionary, making it lightweight and easy to deploy.

For future work, we plan to evaluate the method using different devices, conduct experiments with more users, investigate the construction of a generalized model applicable to multiple users, and evaluate performance using non-random character sequences.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 25K15100.

References

[1] Kai Akamine, Ryotaro Tsuchida, Tsuneo Kato, and Akihiro Tamura. PonDeFlick: A Japanese Text Entry on Smartwatch Commonalizing Flick Operation with Smartphone Interface. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11-16, 2024, Honolulu, HI, USA. ACM. DOI: 10.1145/3613904.3642569

[2] Yusuke Shida, Aki Kobayashi, " Input Character Correction Considering Mis-Touch Probability in Flick Input ", The 76th National Convention of IPSJ, No. 2014, Vol. 1, pp. 51 - 52, 2014. (in Japanese, translated)

[3] Google LLC, ", Gboard - the Google Keyboard," 2025, https://play.google.com/store/apps/details?hl=en_US&id=com.google.android.inputmethod.latin&utm_source=chatgpt.com <Accessed 2025/10/31>

[4] Gary Sivek, Michael Riley, "Spatial Model Personalization in Gboard," Proc. ACM Hum.-Comput. Interact. 6, MHCI, Article 202, 17 pages, 2022. Doi: 10.1145/3546737

Appendix

Fig. 12 shows the error of each consonant of each user.

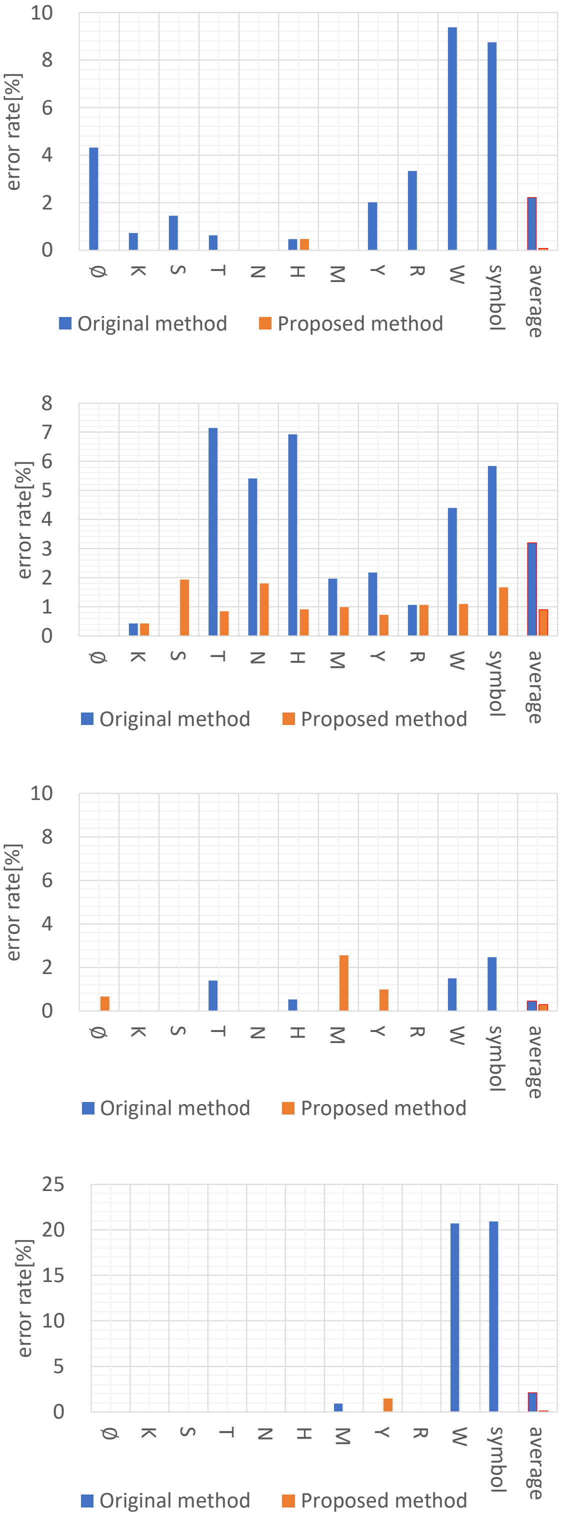

Fig 12. Error date (regions, users)