Probabilistic Adjustment of Dwell Time for Eye Typing

Jimin Pi[1] and Bertram E. SHI[2]
Department of Electronic and Computer Engineering[1,2] and Division of Biomedical Engineering[2]
Hong Kong University of Science and Technology
Kowloon, Hong Kong SAR
jpi@ust.hk[1], eebert@ust.hk[2]

***Abstract*—Requiring a dwell time before selection is a common way to solve "Midas-touch problem" in gaze-based interaction. Choosing the dwell time involves a tradeoff between unintentional selection for short dwell times and slow text entry for long dwell times. We propose a probabilistic model for gaze based selection, which adjusts the dwell time based on the probability of each letter based on the past letters selected. By reformulating the entire problem of gaze-based selection probabilistically, we can naturally integrate the probability of each character naturally and with very few prior assumptions and very few free parameters. It automatically assigns shorter dwell times to more likely characters and longer dwell times to less likely characters. Our experimental results demonstrate that the proposed technique speeds up typing without loss in accuracy. The concept of this can be generalized to other dwell-based applications, leading to more efficient gaze system interaction.**

Keywords – Eye typing, gaze typing, dwell time, eye tracking, human system interaction.

I. INTRODUCTION

Gaze is an important signal for hands-free interaction. For people suffering from motor disabilities, such as ALS and muscular dystrophy, gaze is one of the main channels for them to communicate and interact with the world. With help of eye tracking technology, the user's gaze can be located on a 2D computer screen and used as a control command. Activities of daily living can be controlled through gaze, such as computer interaction[1], wheelchair control [2], control of a robot arm[3].

In gaze-based text entry users input letters by looking at different locations on a virtual keyboard on the screen. The biggest challenge in gaze-based selection is the "Midas-touch problem." Humans use gaze most commonly for purposes other than control, e.g. gathering visual information. Thus, any gaze-based selection cannot assume that the object the user is looking at is the same as the object the user wishes to select, otherwise many objects would be selected unintentionally. There are several ways to deal with this, including adding different behaviors, e.g blinks, to indicate selection or by introducing a dwell time, where users keep their gaze fixed on the object of interest for a fixed period of time, the dwell time, before it is selected [4]. The choice of the dwell time is a tradeoff between speed and accuracy. Short dwell times lead to fast selection, but at the cost of many erroneous selections.

Given the importance of the choice of dwell time, many authors have investigated methods to enable users to select it easily. Spakov and Miniotas proposed to let the user to adjust dwell time on-line based on the exit time [5], the length of time between when a key is selected and when the eyes move away from it. This online adjustment suffers from delayed feedback and uncontrolled variations in the exit time. Marjaranta et al. enabled users to control the dwell time through speed control keys added to a control panel in the interface [6]. This method has the drawback that it requires extra selection time. Interested readers can refer to Räihä and Ovaska's comprehensive analysis of dwell-based gaze typing [7].

The influence of different feedback signals on typing performance was investigated by Majarunta et al. [8]. They showed that visual and audio feedback had a significant impact on typing speed and accuracy; therefore, we also employ feedback in our study.

Some authors have sought to eliminate dwell time altogether. Kristensson and Vertanen investigated the potential of typing a word by looking at the letters in the word without pause [9]. Generally, this kind of system recommends words from a dictionary based on the gaze trajectory. Since the gaze trajectory passes over many unintended letters and may miss desired letters, the gaze trajectory is ambiguous. This system is prone to three types of errors: the extra-letter error, the neighbor-letter error, and the missing-letter error. Filteryedping handles only the extra-letter error [10]. GazeTry can handle all three errors, but is not robust in the presence of multiple errors [11]. Eyeswipe made slight change on the keyboard interface by introducing action button and proposed a "reverse crossing" eye gesture to select start letter and end letter[12]. With start and end letter fixed, word prediction from lexicon is more correct. However, the interface requires extra training to get used to it. Although these techniques can improve typing speed, they have several disadvantages. Pauses between characters within a word, e.g. due to fatigue, can degrade performance dramatically. They also cannot infer words that are not in the dictionary. These are usually dealt with by defaulting back to a dwell-based technique.

This work was sponsored in part by the Hong Kong Research Grants Council under Grant 16211015.

Other techniques to eliminate dwell time, such as Dasher[13, 14], pEYEwrite[15] and EyeWrite [16], use unconventional keyboard layouts, which are re-arranged to facilitate typing based on the likelihoods for the next letter based on previously typed letters. For example, in Dasher, users selected letters by gazing at them as they drift by in boxes whose sizes are proportional to the probability of the letter given the previous selections. Because of their unconventional design, these techniques require additional training, high concentration and may cause additional fatigue. These disadvantages have prevented their widespread adoption.

Researchers have also sought to improve typing speed using letter and word prediction based on previously selected characters. Mackenzie and Zhang used letter prediction to determine which letter the user was fixating at and to determine which letters to highlight as potential next targets, thus reducing search time [17]. However, this may be distracting and cause errors if the highlighted keys do not include the intended key. Many systems present the most likely words given the past inputs in area of the screen outside the keyboard where users can directly select them by fixation [15, 17, 18]. The Augkey system accelerates dwell-based typing by eliminating saccades between the keyboard and the word selection area by showing the prefix and suffixes of the predicted words near the key being focused upon [19]. GazeTheKey system accelerate typing one step further by not only showing the most likely word around the key but also enabling selecting the word through a double-click manner[20].

Very little work has focused on adjusting dwell time based on letter prediction. Intuitively, it makes sense to reduce the dwell time of more likely letters. This might be done in many possible ways. Recently, Mott[21] proposed a heuristic method where the dwell time of more likely keys was reduced depending upon combining information about the likelihood of the letter and the number of other likely letters that are close to it. They also progressively reduced dwell time as the user gets further into the word. Using this method resulted in a 17% gain in average speed. Because it combined multiple heuristics, this method required a large number of parameters, and made a number of ad-hoc assumptions, e.g. that the dwell time should reduce linearly with the probability.

In this paper, we present a more principled way to adjust dwell time based on a Bayesian probabilistic model that allows us to estimate the probability that each key is the desired key, based on the sequence of past gaze points and the sequence of past selected letters. By formulating the problem probabilistically, we obtain a model that has a small number of free parameters and where the assumptions being made are clear and explicit. The adjustment of the dwell time is theoretically well founded, and can also be interpreted intuitively as the accumulation of evidence towards a fixed confidence threshold. More likely keys based on past selections start out with more evidence, and thus require a shorter fixation time before they are selected.

The contribution of this work is two-fold. First, we propose a probabilistic model for gaze based selection, which includes as a special case a standard dwell time based gaze typing system with constant dwell time. Second, we show that this model enables us to incorporate prior information from past selected characters rigorously. The model depends upon only four easily interpretable parameters: one describing the extent to which the user maintains fixation on the desired character, one used to avoid double-selection of the same letter in quick succession, one used to control the dependence on prior context, and a threshold determining the required confidence before a selection is made. Our experiment results demonstrate that the proposed system significantly improves the typing speed and decrease the error rates. Although we focus on improving dwell-based gaze typing, the model is also applicable to other dwell-based selection systems.

II. Method

The probabilistic model consists of two parts. The first is a generative model of the gaze given that a particular letter is being selected. The second is an n-gram based model, which gives the probability that each letter will be selected given the past selections. The two cues are integrated using Bayes rule to give the conditional probabilities that each key is the user's selection target give the past gaze and typing history.

A. Generative Model for Gaze in Dwell-based Typing

Let K be the number of keys on the keyboard and number the keys with integers $\{1, \dots, K\}$. Let $L_i \in \{1, \dots, K\}$ indicate desired input sequence of keys and $M_i \in \{1, \dots, K\}$ indicate the actual input sequence of keys, where i indexes position in the sequence. The keys are arranged in a hexagonal array.

Let $g_1, \dots, g_t$ be a sequence of gaze points after the last selection, where t indexes time and each $g \in R^2$ indicates a gaze location on the screen. We assume that the gaze locations are independent over time, but that the distribution of gaze points changes over time:

$$p(g_1, \dots, g_T | L_i, M_{i-1}) = \prod_{t=1}^{T} p(g_t | L_i, M_{i-1}, t) \qquad (1)$$

The gaze distribution at time index t is given by

$$p(g_t | L_i, M_{i-1}, t) = \gamma(t) q(g_t | L_i) + (1 - \gamma(t)) U_2(g_t, M_{i-1}) \qquad (2)$$

The mixing coefficient $\gamma(t)$ increases linearly from 0 to 1 according to

$$\gamma(t) = \begin{cases} \frac{t \cdot T_s}{T_\mathrm{d}} & \text{if } t \cdot T_\mathrm{s} < T_\mathrm{d} \\ 1 & \text{otherwise} \end{cases} \qquad (3)$$

where T_s indicates the sampling time and T_d = 150ms is the delay before reaching the steady state probability distribution.

The gaze distribution is initially equal to $U_2(g_t, M_{i-1})$, a uniform distribution centered at the previously selected key M_{i-1}. The spatial support of the uniform distribution is square and abuts the spatial support of the distribution for neighboring keys without overlap. The gaze distribution approaches the steady state distribution

$$q(g_t | L_i) = (1 - w) \cdot U_1(g_t) + w \cdot U_2(g_t, L_i) \qquad (4)$$

where $U_1(\cdot)$ is a uniform distribution covering the entire screen area, $U_2(\cdot, L_i)$ is uniform over a square region centered at hypothesized target key L_i, and $w \in [0,1]$ is a mixing coefficient determining the extent to which gaze points near the hypothesized target key are favored.

We adopt this time varying probability distribution to better reflect the actual statistics of the gaze behavior. This choice prevents the problem of double entry of the same key twice in quick succession by explicitly modelling that the eye gaze will still be close to the previously selected key immediately after selection. Other standard dwell time based systems also incorporate a delay before the same key can be selected in order to reduce the problem of double entry [7].

B. N-gram Letter Prediction

We used a within word *n*-gram character model to capture the typing context. To compute the *n*-gram probabilities we utilized the letter sequences frequency counts from [22], which gives counts for word and letter sequences from 23GB of text data in the Google Books N-grams dataset.

With *n*-gram letter prediction, the probability that the user wishes to select each letter depends upon the previous (*n*-1) letters selected. However, we considered only the context within each word (i.e. since the last space character). In other words, for the first character, probabilities were generated by a unigram model computed by maximum-likelihood considering only the first characters of each word of the data set. For the second character, the probabilities were generated by a bigram model computed using maximum likelihood considering only the second character of each word in the dataset, and so on up to the fourth character. In order to deal with missing prefixes (prior context) and for letters in longer words, we also computed a position independent set of uni- to 5-gram models using the Witten-Bell smoothing method using the entire dataset.

The probabilities generated above were further smoothed by mixing them with a uniform probability distribution over all keys. Denoting the n-gram probability computed as described in the previous paragraph by P_n, and $M_{i-4}^{i-1} = (M_{i-4}, \dots, M_{i-1})$ to be the last four selected characters, we computed

$$p(L_i|M_{i-4}^{i-1}) = \lambda * P_n + (1-\lambda) * \frac{1}{K} \tag{5}$$

where the parameter λ controlled the amount of smoothing. We used this to avoid zero probabilities, which might occur for unseen letter combinations in the data set, and to avoid probabilities which are too close to one, which might lead to automatic selection of a character. When $\lambda = 0$, all keys are considered to be uniformly probable, essentially ignoring all past context.

C. Bayesian Cue Integration

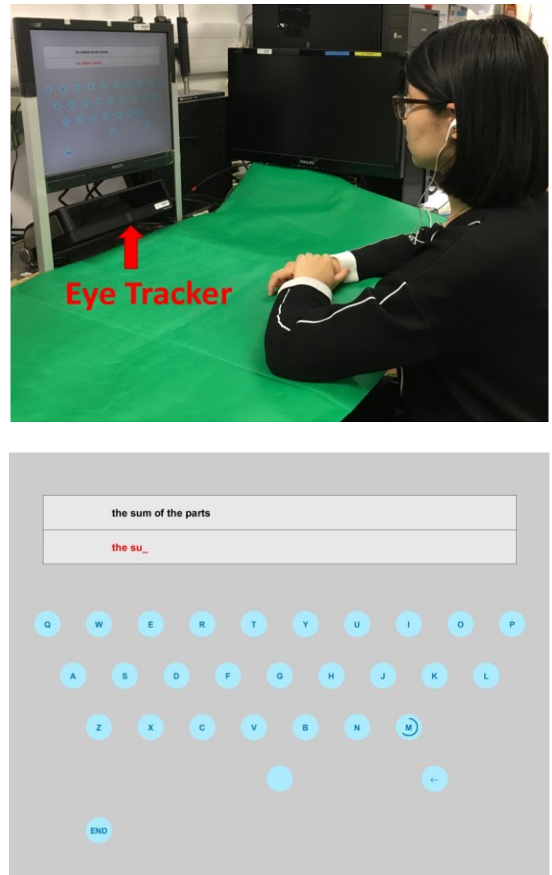
Let L_i indicate the current key the user wishes to select, M_{i-1}^{i-4} indicate the past four letters actually selected, and g_1^t denote the sequence of gaze points up to time *t* after the last selected character, M_{i-1}. Note that each selection is based on multiple gaze points. Using Bayes rule, we find the posterior probability

$$p(L_i|g_1^t, M_{i-4}^{i-1}) = \frac{p(g_1^t|L_i, M_{i-1}) * p(L_i|M_{i-4}^{i-1})}{\sum_{k=1}^{K} p(g_1^t|L=k, M_{i-1}) * p(L=k|M_{i-4}^{i-1})} \tag{6}$$

Selection is performed by comparing the value of $p(L_i|g_1^t, M_{i-4}^{i-1})$ with a pre-defined threshold α.

Intuitively, at the beginning of the selection period, due to the incorporation of the prior context, the posterior probability is already closer to the threshold for characters that are more

(a)



(b)

Figure 1. (a) The experimental setup. (b) The keyboard interface. The closing circle indicates the progress of the selection.

likely. Although the user's eyes are likely still at the last key selected, this does not initially change the posterior probability of the keys by much, since the gaze likelihoods for all keys are similar due to the time delay in equation (2). As the user seeks the next key, the gaze position becomes weighted more and more, increasing the posterior probability of the keys being looked at. Eventually, if the user spends long enough looking at a particular key, the posterior probability rises above the threshold and the key is selected. The amount of time required for this to happen is the dwell time, which will be shorter for more likely keys due to their higher initial posterior probability.

To obtain a system that is very similar to standard dwell time based typing, we set the smoothing parameter $\lambda = 0$, which eliminates the contribution of prior context. Assuming that the delay $T_d = 0$, the posterior probability is given by

$$p(L_i|g_1^t, M_{i-4}^{i-1}) = \frac{q(g_1^t|L_i)}{\sum_{k=1}^{K} q(g_1^t|L=k)} \tag{7}$$

where $q(g_t|L_i)$ is given in equation (2). Assuming that the user is looking within the box corresponding to the desired key for the entire selection period, this can be expressed as

$$p(L_i|g_1^t, M_{i-4}^{i-1}) = \frac{1}{1+(K-1)\left(1+\frac{w}{1-w}\frac{A_S}{A_K}\right)^{-t}} \tag{8}$$

where A_S and A_K are the area of the screen and the area associated with the key. This probability increases monotonically from $1/K$ when $t = 0$ to a maximum value of 1 as t approaches infinity. The amount of time it takes to reach the threshold (the dwell time) increases with α and decreases with w.

For given values of α and w, we define the *nominal dwell time* of the probabilistic system to be $t \cdot T_s$ where t is the time index at which $p(L_i|g_1^t, M_{i-4}^{i-1})$ in equation (8) exceeds the threshold the α and T_s is the sampling period. When $w = 0$, the gaze likelihood is the same for all keys; the gaze is uninformative about the user intent; and the selection is never made, i.e. the nominal dwell time is infinite. When $w = 1$, the gaze is assumed to a perfect indicator of intent, and selection occurs immediately based on the first gaze point, i.e. the nominal dwell time is T_s.

In our experiments (data not shown), we observed little difference in performance between a standard implementation of dwell-based typing with a 150ms delay to avoid double-entry and our probabilistic system with $\lambda = 0$ and $T_{\mathrm{d}} =$ 150ms. Because of our probabilistic treatment of gaze, the system handles non-ideal behavior, e.g. short intermit drifts away from the selection square naturally gracefully, without requiring ad-hoc measures commonly used in other dwell based methods, e.g. looking for a certain number of gaze points at the same key over a larger window of time.

III. Experimental Procedure

A. Participants

Altogether 10 participants (6 males and 4 females, average age of 26.1, SD=1.45) took part in the experiments. All had normal or corrected-to-normal vision. Two had prior experience in using eye trackers. None had experience in gaze typing.

B. Experimental setup

The eye tracker used was Tobii X60 with sampling rate of 60HZ. The subject is seated in front of a 19 inch 1280×1024 resolution monitor at distance around 60cm (Figure 1(a)). A chinrest was used to help avoid large head movements. Standard nine-point eye tracker calibration was performed before the experiment.

The interface we used was shown in Figure 1(b). Keys are arranged in the same way as the QWERTY keyboard layout. The horizontal distance between neighboring keys and vertical distance between rows are both 120 pixels. The space key is placed in the middle and below the letter keys, with the backspace key at its right. We also place an "END" key to indicate the finish of current sentence entry.

The stimulus sentence is placed on the top panel, in black. The transcribed string is in red, placed below the stimulus. We did not include any other symbols or function keys in the experiment. Typing was case insensitive. All keys had the same sized active selection area, a 120×120 pixel square centered at the key center.

We also included visual and auditory feedback in the experiment. An animated circle closing around the letter is shown to indicate the progress towards key selection. The selection was made when the circle closed. After selection, the key flashed in red and a 'click' sound was generated.

TABLE I. w VALUES USED IN THE EXPERIMENT

Dwell time	*500ms*	*400ms*	*300ms*	*200ms*
w	0.0022	0.0028	0.0040	0.0065

C. Procedure

We compared a standard (non-adaptive) dwell-based gaze typing system and our proposed probabilistic gaze typing system, where dwell time adapts according to the prior selection history. For both the standard and probabilistic models, the selection of 'Backspace' and 'END' key followed the standard dwell-based technique.

In the standard dwell-based gaze typing system, keys were selected if the user fixated on their associated selection box continuously for the dwell time. However, for the previously selected key, the dwell time clock could be started only 150ms after the previous selection to avoid double entry.

For the probabilistic gaze typing system, we set the threshold α = 0.9, and chose four values of w according to equation (8) so that the threshold was exceeded after nominal dwell times of 500ms, 400ms, 300ms and 200ms. These values of w are listed in Table 1. We set T_{d} = 150ms to avoid double entry. We fixed the value of the smoothing parameter $\lambda =$ 0.75 for all experiments to ensure consistency among the experimental results. During actual usage, we expect that λ could be chosen by the user according to their preferred amount of dwell time adaptation.

For each subject, the experiment had two phases: a learning phase and an evaluation phase. The goal of the learning phase is to familiarize subjects with the interface and gaze typing. Subjects conducted 8 sessions of gaze typing: four sessions of standard dwell-based typing where the nominal dwell-time varied between 500ms, 400ms, 300ms and 200ms and another four sessions of probabilistic dwell-based typing with same nominal dwell times. During each session, subjects typed three phrases, chosen from the phrase sets for evaluating text entry techniques in [23].

In the evaluation phase, each subject also conducted 8 sessions, alternating between the standard and probabilistic models with the same nominal dwell time. Which technique was used first was randomly selected at the beginning of the evaluation phase, and remained the same for all sessions. During the evaluation stage, each subject typed 15 phrases, which were distinct from those used in training, but chosen from the same database. To allow time for adaptation, we discarded the results from the first phrase in each session. Subjects were instructed to correct errors only when they noticed them immediately. Subjects took 10 minute breaks after every two sessions.

Subjective evaluations were collected after each session in advanced phase. NASA TLX workload scale[24] is used to evaluate the task workload. The workload is a combined evaluation from 6 aspects: mental demand, physical demand, temporal demand, performance, effort and frustration. We also asked the subject to rate each session in terms of speed, accuracy and overall performance in 10-point scale.

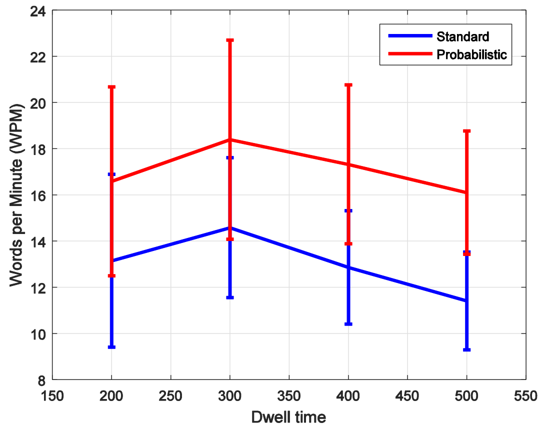


Figure 2. Words per Minute (WPM). Error bars indicate the standard deviation.

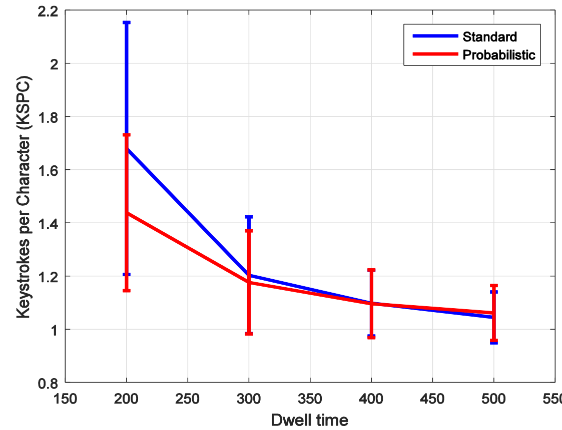


Figure 3. Keystrokes per Character (KSPC). Error bars indicate the standard deviation.

IV. Experimental Results

We adopted several metrics to evaluate results in terms of both overall performance and detailed performance.

A. *Words per Minute (WPM)*

Words per minute (WPM) has long been a standard metric to measure text-entry speed. Each 'word' corresponds to five characters. The WPM is obtained by dividing the number of words entered by the number of minutes required to enter the text. The time for entering 'END' key was not counted. This metric measures the overall typing speed performance. Both long selection time per character and corrections due to a large number of selection errors lead to low WPM values.

Figure 2 shows the mean and standard deviation of WPM obtained for different nominal dwell times averaged across all subjects and all phrases. The probabilistic technique leads to significant increases in typing speed. We conducted two-way analysis of variance where each phrase was considered as a replication to analyze the impact of Technique×Dwell time. There was a significant main effect for technique ($F_{1,1112} = 422.63$, $p < 0.0001$) and for dwell time ($F_{3,1112} = 31.87$, $p < 0.0001$). There was not a significant effect for Technique×Dwell time interaction ($F_{3,1112} = 2.1$, $p = 0.0991$).

The fastest typing was obtained when the nominal dwell time was set to 300ms for both the standard and probabilistic systems. At this setting, the increase in speed for the probabilistic system was significant ($p < 0.05$). The average speed of the fastest probabilistic system is 26% faster than that of the fastest standard dwell-based system. This increase is much larger than that observed in prior work that adjusted dwell time dynamically (17%) [21]. The fastest speed of our system (18.4 WPM) is also larger (12.39 WPM).

B. *Keystrokes per Character (KSPC)*

In addition to speed, we also care about the accuracy. Generally, number of corrections required increases as the dwell-time decreases. The keystrokes per character (KSPC) measures the how many corrections are made during text entry. It is defined as the ratio between the number of real keystrokes and the minimum number of keystrokes needed to enter a string. KSPC = 1 indicates no corrections. Any corrections during text entry will make KSPC larger than 1.

Figure 3 shows the mean and standard deviation of the KSPC obtained at different nominal dwell times. The performance of the standard and probabilistic systems is similar when the nominal dwell time is long. Both have low correction rates. However, for shorter nominal dwell times, the probabilistic model gives a lower error rate. The improvement at the shortest nominal dwell time setting (200ms) is significant ($p < 0.05$). Thus, the probabilistic system introduces fewer errors than the standard system. Our approach not only facilitates the selection of highly probable letters by shortening their dwell time, but also lengthens the dwell time for unlikely letters. This avoids incorrect selections at short nominal dwell-times.

C. *Milliseconds per Character (MSPC)*

In addition to measures of word-level performance using the WPM and KSPC, we also computed character-level performance. The milliseconds per character (MSPC) measures the average time for selecting one character. Unlike the previous metrics, we do data cleaning at this stage, since we hope to investigate error free performance under each setting. We adopted the same data cleaning process as in [7]. Only selections that are entered correctly after another correct character were considered. Selections were also discarded if the subject glanced at the string line during the process of selection.

Figure 4 shows the mean and standard deviation of MSPC obtained in each session. The proposed model used less time on average for single selection, especially for long dwell time case. This indicate that the n-gram letter prediction works well. It successfully decreases the dwell time of the target key for most of the selections.

D. *Total Error Rate*

MSPC is intended to measure the error-free typing speed. Since there is a tradeoff between accuracy and speed in dwell-based typing, we computed the total error rate as a character level evaluation considering both speed and accuracy. Although KSPC measures the overall occurrence of corrections, there are also errors that are not corrected. These errors remains in the transcribed text, which is actually typed

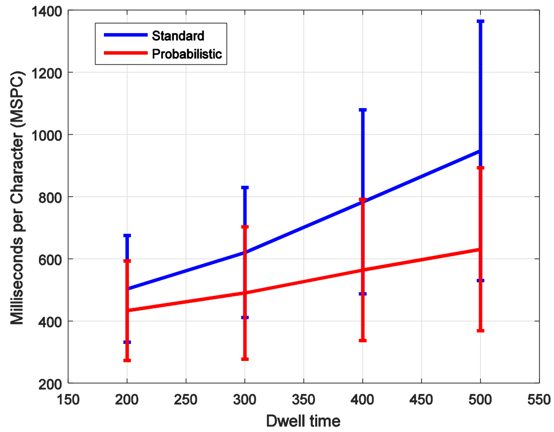


Figure 4. Milliseconds per Character (MSPC). Error bars indicate the standard deviation.

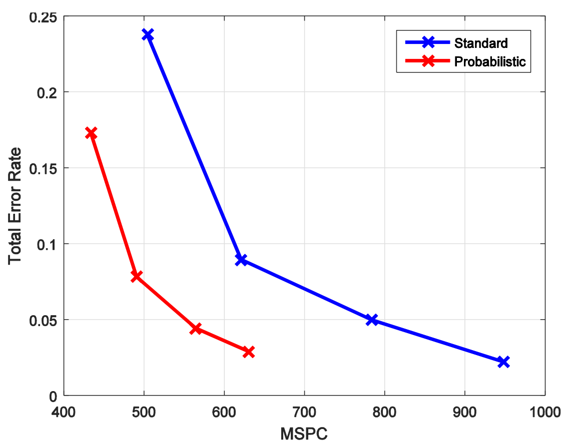


Figure 5. Relationship between the total error rate and the MSPC.

by the subjects. To take these into account, Soukoreff proposed a metric of total error rate, which accounts for both corrected errors and uncorrected errors [25]. The total error rate is defined as the ratio of incorrect character entries to the total character entries. Incorrect character entries include both those that were fixed and those that were not. Unfixed errors were identified by the minimum number of editing primitives (insertion, deletion, and substitution) required to transform the transcribed string to the target string.

Figure 5 shows the relationship between total error rate and the MSPC. The red curve, corresponding to the proposed method is generally below the blue curve of the standard dwell-based technique. This demonstrates that our system functions at faster speed with lower error rates.

E. Subjective Evaluation

Figure 6 shows the subjective evaluation of workload from NASA TLX averaged over subjects for each nominal dwell time. In all cases, the probabilistic system requires less perceived workload than the standard system.

We also asked subjects to rate each session in terms of speed, accuracy and overall performance. Each session is scored using a 10-point scale, where 10 stands for fastest, most accurate or best overall. Table II shows the mean and standard

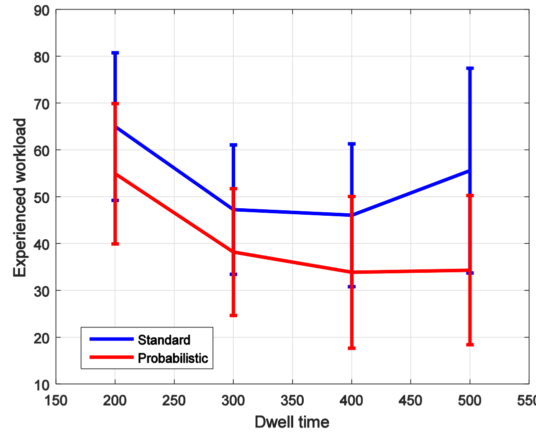


Figure 6. Subjectively experienced workload.

deviation of these ratings for each experimental condition. The subjective evaluations of speed match the trend in MSPC. The subjective evaluation of accuracy matches the trend in KSPC. In the overall evaluation, the proposed method generally achieves higher score than standard dwell-based technique. The probabilistic technique with nominal dwell time equal to 300ms achieves highest score in overall evaluation.

V. Conclusion

We have described a probabilistic model of dwell-based gaze typing. Using this model, we have constructed a gaze typing system where the dwell time assigned to different keys varies according to the past typing history. Unlike prior approaches to adapting dwell time, which were based on heuristics and required manual tuning of a large number of parameters, our technique is mathematically rigorous and is based on a small number of parameters, each of which has an intuitive interpretation. Our experimental results showed that the proposed method improves the speed at the same fixed accuracy, and improves accuracy at the same nominal dwell time.

TABLE II. Subjective Score

		500ms	***400ms***	***300ms***	***200ms***
Speed	**Std.**	**3.2±2.44**	**4.9±1.66**	**6.2±1.87**	**8.8±0.92**
	Prob.	**5.7±2.00**	**7.2±1.23**	**8.1±0.99**	**9.5±0.71**
Acc.	**Std.**	**8.0±2.40**	**7.9±2.13**	**6.9±1.52**	**4.3±2.79**
	Prob.	**9.0±0.82**	**8.4±1.43**	**7.4±1.35**	**5.5±2.37**
Overall	**Std**	**4.2±2.90**	**5.9±1.37**	**6.6±1.17**	**5.1±2.38**
	Prob.	**6.4±1.78**	**7.8±1.23**	**8.2±1.23**	**5.9±1.66**

References

[1] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey, "Human-computer interaction using eye-gaze input," *IEEE Transactions on systems, man, and cybernetics,* vol. 19, no. 6, pp. 1527-1534, 1989.

[2] C.-S. Lin, C. Ho, W. Chen, C. Chiu, and M. Yeh, "Powered wheelchair controlled by eye-tracking system," *Optica Applicata,* vol. 36, no. 2/3, p. 401, 2006.

[3] S. Dziemian, W. W. Abbott, and A. A. Faisal, "Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing & drawing," in *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference on*, 2016, pp. 1277-1282: IEEE.

[4] R. J. Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Transactions on Information Systems (TOIS),* vol. 9, no. 2, pp. 152-169, 1991.

[5] O. Špakov and D. Miniotas, "On-line adjustment of dwell time for target selection by gaze," in *Proceedings of the third Nordic conference on Human-computer interaction*, 2004, pp. 203-206: ACM.

[6] P. Majaranta, U.-K. Ahola, and O. Špakov, "Fast gaze typing with an adjustable dwell time," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 357-360: ACM.

[7] K.-J. Räihä and S. Ovaska, "An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 3001-3010: ACM.

[8] P. Majaranta, I. S. MacKenzie, A. Aula, and K.-J. Räihä, "Auditory and visual feedback during eye typing," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA, 2003, pp. 766-767, 765979: ACM.

[9] P. O. Kristensson and K. Vertanen, "The potential of dwell-free eye-typing for fast assistive gaze communication," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, Santa Barbara, California, 2012, pp. 241-244, 2168605: ACM.

[10] D. Pedrosa, M. D. G. Pimentel, A. Wright, and K. N. Truong, "Filteryedping: design challenges and user peformance of dwell-free eye typing," *ACM Transactions on Accessible Computing,* vol. 6, no. 1, pp. 1-37, 2015.

[11] Y. Liu, C. Zhang, C. Lee, B.-S. Lee, and A. Q. Chen, "GazeTry: swipe text typing using gaze," in *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, Parkville, VIC, Australia, 2015, pp. 192-196, 2838804: ACM.

[12] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke, "EyeSwipe: Dwell-free text entry using gaze paths," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, Santa Clara, California, USA, 2016, pp. 1952-1956, 2858335: ACM.

[13] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, "Dasher - a data entry interface using continuous gestures and language models," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, San Diego, California, USA, 2000, pp. 129-137, 354427: ACM.

[14] O. Tuisku *et al.*, "Now Dasher! Dash away!: longitudinal study of fast text entry by eye gaze," in *Proceedings of the 2008 symposium on Eye tracking research & applications*, Savannah, Georgia, 2008, pp. 19-26, 1344476: ACM.

[15] M. H. Urbina and A. Huckauf, "Alternatives to single character entry and dwell time selection on eye typing," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, Austin, Texas, 2010, pp. 315-322, 1743738: ACM.

[16] J. O. Wobbrock, J. Rubinstein, M. W. Sawyer, and A. T. Duchowski, "Longitudinal evaluation of discrete consecutive gaze gestures for text entry," in *Proceedings of the 2008 symposium on Eye tracking research & applications*, Savannah, Georgia, 2008, pp. 11-18, 1344475: ACM.

[17] I. S. MacKenzie and X. Zhang, "Eye typing using word and letter prediction and a fixation algorithm," in *Proceedings of the 2008 symposium on Eye tracking research & applications*, Savannah, Georgia, 2008, pp. 55-58, 1344484: ACM.

[18] P. Majaranta, "Text entry by eye gaze," Ph.D. dissertation. Department of Computer Science, University of Tampere, Tampere, Finland. 2009.

[19] A. Diaz-Tula and C. H. Morimoto, "AugKey: increasing foveal throughput in eye typing with augmented keys," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, Santa Clara, California, USA, 2016, pp. 3533-3544, 2858517: ACM.

[20] K. Sengupta, R. Menges, C. Kumar, and S. Staab, "GazeTheKey: Interactive keys to integrate word predictions for gaze-based text entry," presented at the Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion, Limassol, Cyprus, 2017.

[21] M. E. Mott, S. Williams, J. O. Wobbrock, and M. R. Morris, "Improving dwell-based gaze typing with dynamic, cascading dwell times," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2017: ACM.

[22] P. Norvig, "English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU," http://norvig.com/mayzner.html

[23] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA, 2003, pp. 754-755, 765971: ACM.

[24] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): results of empirical and theoretical research," *Advances in Psychology,* vol. 52, pp. 139-183, 1988/01/01 1988.

[25] R. W. Soukoreff and I. S. MacKenzie, "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*, Ft. Lauderdale, Florida, USA, 2003, pp. 113-120, 642632: ACM.