IEEE Proceedings of 4^{th} International Conference on Intelligent Human Computer Interaction, Kharagpur, India, December 27-29, 2012

EyeBoard: A Fast and Accurate Eye Gaze-Based Text Entry System

Prateek Panwar[1†], Sayan Sarcar[2*] and Debasis Samanta[3*]
[*]Indian Institute of Technology Kharagpur, Kharagpur, India - 721302
[†]Sri Dharmasthala Manjunatheshwara College of Engineering & Technology, Karnataka, India - 580002
Email: [1]prateekpanwar31@gmail.com, [2]mailtosayan@gmail.com, [3]debasis.samanta.iitkgp@gmail.com

***Abstract*—Over the last three decades, eye gaze has become an important modality of text entry in large and small display digital devices covering people with disabilities beside the able-bodied. Despite of many tools being developed, issues like dwell time optimization, visual search time and interface area minimization, eye-controlled mouse movement stability etc. are still points of concern in making any gaze typing interface more user friendly, accurate and robust. In this paper, we propose *EyeBoard*, an gaze-based text entry system which optimizes the constituent virtual keyboard layout with respect to eye gaze movement as well as adjusts dwell time dynamically in accordance with user comfort level. Performance evaluation shows that proposed interface achieves on an average 14% higher text entry rate over the existing interfaces. As designed, the proposed interface can effortlessly be suited in small display devices like Mobile phone, PDA etc. Although the experiments have been conducted with able-bodied users, those can easily be replicated for people having severe motor disabilities.**

***Index Terms*—Eye gaze-based text entry mechanism, user interface design and evaluation, human factors and ergonomics**

I. INTRODUCTION

Enormous growth in Information and Communication Technology (ICT) causes indiscriminate availability of digital devices like Cell Phone, iPhone, Tablet PC, iPad etc. along with Desktop PC and Laptop. In this context, many alternate interaction methods have been evolved to facilitate speedy as well as accurate text entry. Among many alternatives, gaze controlled interaction has strongly been evolved in last few decades [1], and lot of effort has been put for developing several applications and interfaces. This mechanism is having moderate similarity with any traditional text entry methods; only difference is that the instead of hand, the controlling human organ is eye. Also, eye gaze-based text entry is among those which, in addition to able-bodied people, extends the applicability to disabled, capable of visual interaction and having good vision. Many applications support text entry through eye gaze ([2], [3], [4]), even in the mobile environment [5].

The eye gaze-based text entry, often called as *eye typing*, is accomplished by direct pointing or looking at the desired letter within interface [6]. To type by gaze, typical computational requirements include an on-screen keyboard and an eye tracking device. Selection of a letter, i.e. *eye press*, is accomplished by hovering on the letter for a slightly prolonged duration, i.e. *dwell time*. Many gaze typing systems support *eye blink* besides eye movement during typing. As the gaze-based text entry mechanism is gaining popularity among alternate text entry mechanisms, it brings a number of design issues that make gaze-based text entry a unique technique with own set of research problem.

A well known problem of dwell-based gaze input is *Midas Touch* problem [7] which can lead to a situation where commands are activated unintentionally, while users are scanning the interface for their interested information. This problem hinders the development of gaze-based interactions. Attempts have been made to overcome *Midas touch* problem [8]. Decreasing the number of keys and space between keys can be used to save screen space [3]. In contrast, bigger size of the keys can make the typing task more easier, even in a setup with low spatial resolution [9]. Thus, in some cases, instead having fewer keys, the interface takes larger screen space. So, to optimize between eye movement, screen space and user comfort, an optimal size of the keys and space between keys need to be decided.

While typing, visually searching the next character, is significantly affected by the features of the interface such as color, orientation, shape, size, spatial frequency, etc. [10]. Optimizing screen area is one important constraint, specially in developing system for small display devices. Špakov and Miniotas [11] developed a keyboard that saves screen space as well as instantly usable and not require any special learning. They initially took a keyboard layout which is already familiar to the user (such as QWERTY) and showed a part of the keyboard to save screen space, called as *scrollable keyboards* [11].

A typical issue in dwell-based eye pointing is that, how long is long enough to maintain speed-accuracy trade-off at different levels of cognitive complexity. A long dwell time is good for preventing false selections but a long fixation on the same target can be tiring for eyes. In contrast, shorter dwell time enhances the chance of *Midas Touch* problem. So, we cannot conclude that the shorter the dwell time, the better text entry rate. The dwell time also sets a limit for the maximum typing speed as the user has to wait for the dwell time to elapse before each selection. Majaranta and Räihä [6], state the fact that most gaze typing evaluations were conducted with novices using a constant, fairly long dwell time ($450 - 1000$ ms). Recently, Wobbrock et al. [12] used a short dwell time of 330 ms and achieved text entry rate of 7 wpm. Špakov and Miniotas [13], and Majaranta and Räihä [6] studied automatic

978-1-4673-4369-5/12/$31.00 ©2012 IEEE

adjustment of dwell time. Although the typing result of those systems were better, they are suffering in *delay* and *involuntary variation* problems. Therefore, a trade-off exists among dwell time, text entry rate and accuracy of the interface.

Diminishing dwell time is another avenue to increase eye typing rate. Experimental analysis on eye typing based interface task reveals that for a user, throughout the experiment, dwell time may vary depending on his comfort level. Keeping this in mind, Majaranta et al. [14] developed an interface which dynamically adjust dwell time during experiment. They got promising result in case of increasing eye typing rate. Urbina and Huckoff [4], Kristensson and Vertanen [15] proposed a research direction for eye-typing which is potentially much faster as well as dwell-free.

In this paper, we are trying to develop a gaze-based text entry interface fulfilling the following objectives.

1) Optimizing the virtual keyboard layout with respect to size of the key, space between keys and zoom of the layout to achieve minimum eye movement while composing text.
2) Dynamically adapting the dwell time of the interface based on typing efficiency of the users.

The rest of the paper is organized as follows. The design methodology of the interface is presented in Section II. Section III discusses the overall setup established for the experiments. Experimental results and analysis of our proposed system are presented in Section IV. Finally, Section V concludes the paper.

II. METHODOLOGY

We develop a gaze-based text entry system to cater the above mentioned issues. The proposed system is named as *EyeBoard*, which consists of a keyboard supporting *Eye typing*. *EyeBoard* provides a keyboard layout optimized against key size, space between keys and zooming of the layout with respect to minimum eye movement. Side by side, the system is capable to adapt the dwell time dynamically, depending on the user's comfort level during text typing task.

A. Selecting the effective on-screen keyboard

We have studied several existing design principles of virtual keyboards in English namely *Dvorak*, *Opti*, *Fitaly* and *Lewis*. *Dvorak*, also called a *Simplified Keyboard*, is different with the *Qwerty* design through an analysis of the relative frequency of used letters [16]. The layout attempts to minimize hand and finger movements with an emphasis on the right hand. The *Fitaly One-Finger* keyboard is designed to optimize mouse movements during the text entry with one finger, stylus or pen [16]. In the layout, two spacebars are present keeping proximity of the most common letters in English (e.g. E, T, A, H). *Opti*, proposed by Mackenzie and zhang, is the optimized virtual keyboard layout in English [16]. The optimization has been done with respect to typing speed using trial and error method, Fitts' law and bi-gram frequency of characters. Lewis et al. proposed a 5×6 virtual keyboard where keys are placed alphabetically in the matrix following row-major order [16].

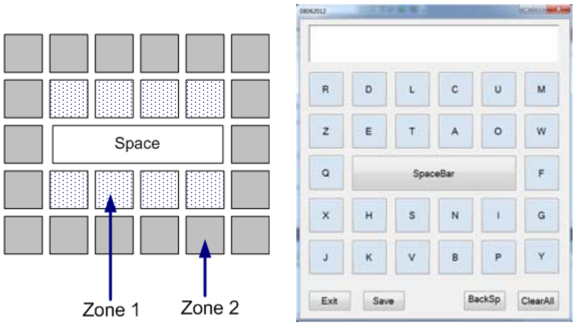
To understand the impact of these keyboard design principles in performing gaze-based text entry task and analyze the human acceptance of the layouts, we perform expert user-based evaluation [17] with our eye tracking setup. The evaluations have been conducted by 5 to 6 expert participants (UG/PG students) having previous experience in participating user evaluation. The details about the participants, apparatus, texts to be typed and evaluation procedures are mentioned in Section III. During the experiments, we keep the keyboard dimensions same like both key width and height as 70 pixels, horizontal distance between 2 keys is 8 pixels and zoom is 30% of original size. For each session, the details about the task performing along with the calculated result are stored in a log file. We observe the participants' behavior during experiments and conclude that eye movements during experiments are more in two cases, more spacebars present in the interface (many users cannot manage selecting a spacebar in close proximity rather habituated using a fixed spacebar), and keys are arranged alphabetically rather than frequency-based. This discussion concludes as the gaze movement during text entry is less in *Fitaly* than other designs. As a consequence, we further continue with *Fitaly* keyboard (accommodating 27 keys in a 6×5 matrix) design.

B. Minimizing the eye movement

Fitaly, a frequency based keyboard layout contains two spacebars. Here characters are arranged according to their unigram frequencies in rows. After experimenting more on *Fitaly* layout with different standard sentences, we find that user's eye always moves around the middle portion of the layout while searching for next character to be typed. Also, searching for next character is always started from the centre and spread around. Moreover, there is no need of two spacebars because more than one spacebar causes more eye gaze movement.

Taking those design implications into account, the *Fitaly* layout is modified and included in the proposed interface. Here, we can reduce the search time, if frequently occurring characters are placed on or around the central region of the keyboard. If it is possible to accommodate all frequently occurred characters in central region, then the characters may be place into two zones, that is, the most frequently occurring characters in the central zone (zone 1) and next most frequently occurring characters in the outer zone surrounding to centre zone (zone 2). We use Mayzner and Tresselt's table [18] which provides unigram as well as bigram frequencies of English characters. Initially, we set the space character at the middle of the central zone (as space is the highest probable character among any English characters) following most frequently occurred characters around it (zone 1) and other less probable characters can be spatially arranged in other zone (zone 2) depending on their frequency of occurrence (Fig. 1(a)). To support user further, each character, on hovering, is being spoken out for better understandability.

After modifying the existing *Fitaly* keyboard, a requirement has become obvious which leads to a solution of three constraints: value or range of size of the key, distance between



(a) Multi-zonal layout for virtual keyboard

(b) Interface with Modified Fitaly layout

Fig. 1. Schematic and implemented diagram of modified FITALY keyboard

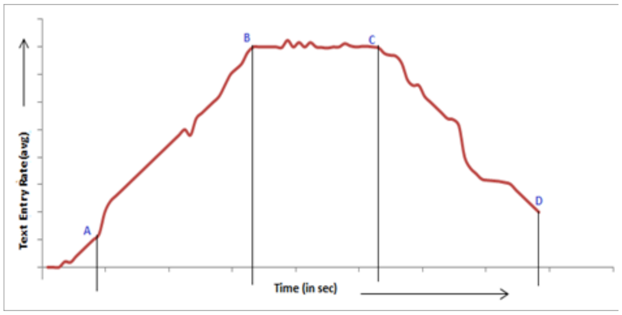


Fig. 2. Observation of Dwell time in a pilot study

keys and zoom of the keyboard. This mechanism fixes the optimal layout with the features which, in future, takes care of users as well as HCI designers in terms of comfort in eye movements (for users) and specifying minimum screen occupancy (for designers) of the interface. It may also be noted that, we are not proposing a common algorithm which works for any keyboard (where orientation, number of characters per row and column are different), rather providing a way of solving the problem. While finding the solution, first we increase three features (size, distance and zoom of the keyboard) of the *Fitaly* keyboard programmatically to cover the maximum area of the display. The participants are requested to compose $2-3$ sentences for a particular session. Further, three feature values are decreased a unit each and experiments are repeated with same users. If text entry rate increases, then next session starts with decremented feature values. However, if consecutive $3-4$ sessions produce negative result i.e. lower text entry rate, then the lowest feature values which yield highest text entry rate are kept. The *Fitaly* keyboard, with lowest feature values achieving highest text entry rate is depicted in Fig. 1(b).

C. Dynamically adapting dwell time while composing text

According to literature, the normal duration of dwell time varies from $500-1000$ms. It is found that long dwell time may also be tiring for eyes and hinders concentration; too short dwell time also increases the *Midas touch* problem. Moreover, dwell time is one of the most important parameter which directly or indirectly affects the text entry rate. So there exists a range in dwell time for which gaze typing becomes comfortable and accurate. After taking experiments on different users, we analyze the results and observe that every user follows a certain tendency while gaze typing; pilot study result, Fig. 2.

Initially, when the experiment is started, user takes moderate time to be familiar and interact with the given layout. So, the text entry rate is picked up gradually. Then, after moderate time, user starts gaining confidence in accessing the interface and text entry rate increases evenly. The growth of text entry rate saturates over time. Over the time, user gradually feels tiring up in continuous eye movement for entering text in a gaze-based interface. Also, when user gets frustrated, involuntary head or eye movement is occurred and thus calibration gets disturbed. As a result, text entry rate decreases gradually over time.

Therefore, by observing above scenario, we try to help user in such a way that the saturation stage of text entry rate remains maximum time throughout gaze typing and the span increases over the time. We introduce the concept of dwell time adaptation to achieve the earlier. Thus, if user's typing rate is increasing gradually, then decrement of dwell time helps for further increment of typing rate. On the other hand, if users are getting exhausted or stressed, dwell time is increased gradually to avoid improper concentration on the intended key. Our proposed interface analyzes the data after every 1 second. If the text entry rate is continuously increasing, dwell time is automatically decreased by 0.1 second. But if the user feels uncomfortable (commits error etc.) with the change, then dwell time is automatically increased (Fig. 3 depicts the scenario).


A QUICK B

A QUICK BROWN FOX JUMPS OVER THE LAZY DO

Dwell Time : 10

Dwell Time : 9

(a) Interface with dwell time 1 second, initially which has been set

(b) Depending on the performance of participant, it becomes 900 milliseconds

Fig. 3. Interface dynamically adjusting dwell time

When dwell time is less than or equal to 700ms, the sound effect is stopped to support better concentration as providing sound may be redundant and may reduce attention at lower response time.

D. Apparatus

All experiments are conducted using 2.2GHz Intel Core2Duo processor with 15″ wide screen LCD color monitor having 1440×900 resolution. Modified *Sony PlayStation*

Eye webcam, original lens is replaced by manual focus and Infrared (IR) filter removed lens, *IR Lamp*, consisting a matrix of 10 IR LED, along with open source *ITU GazeTracker* software, developed by IT University of Copenhagen, are used for experiments. The developed keyboard interfaces for experiments is written in $C\#$ using Visual Studio 2010, which can be accessed through mouse or single pointer-based touch screen input. The key press events are recorded automatically and stored in a log file using a separate event hooking program. Another window hook program is developed to track gaze positions; also written in $C\#$. All experiments are performed in Windows 7 environment. Controlled light conditions and positioning of the setup are maintained.

E. Participants

Eight participants (5 male, 3 female) are voluntary recruited from the local university campus. Participants ranged from 22 to 33 years (mean = 25). All are daily computer users, access on an average 7 hours per day, but no prior experience with eye tracking. All participants, except one with contact lenses, have normal vision and expertise in composing text through digital devices. 6 participants are right-eye dominant and 2 are left-eye dominant, as determined using an eye dominance test [19].

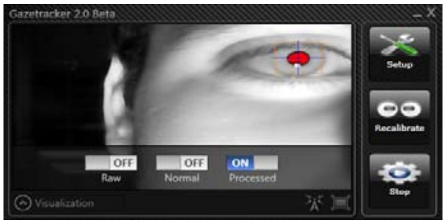
F. Designs

Four designs including *EyeBoard* are chosen for experiments. The designs are namely, optimized *scrollable* keyboard which saves the screen space proposed by Špakov et al. [11] (Design 1), keyboard designed by Majaranta et al. [14] maintaining less dwell time (Design 2) and a design called *Iwrite*, which is a square shaped interface keeping characters at outer side and text area in middle for gaze typing [4] (Design 3). In the experiments, 9 texts are considered for typing. Each text contains 10 phrases containing approximately 25 characters. The selected phases are easy to remember. The phrase set is tested for its correlation with common English using the frequency counts in Mayzner and Tresselt's corpus[18]. The result is $r = 0.973$ for the single-letter correlation and $r = 0.908$ for the digraph correlation. Each participant performed 3 sessions, each for a corpus, for each of the 4 keyboards.

G. Procedure

To perform user-based evaluation, users first need to synchronize their eye movement with the gaze tracker. The synchronization process, calibration, with detail description is provided next. The typing session starts after calibration, when mouse pointer moves with eye gaze. Participants are asked to remember the phrases because experiments, once started, could not be stopped as it may require recalibration. The text can be presented on top of the interface. But, this type of arrangement increases the overall mouse movement and as a result, text composing rate gets slowed down [20]. The participants are further instructed to transcribe the phrases as fast as possible while making as few errors as possible. Correcting errors is possible by erasing text using backspace key and retyping it.

Calibration: Before an eye tracking system can work with eye gaze, it must be calibrated for the specific user. This is usually done by showing a few (usually nine equally spaced, one at a time) points on the screen and asking the user to gaze at the points. The images of eye are stored separately for each point, and analyzed with corresponding screen coordinates of the same point. These main points are used to calculate any other point on screen via interpolation of the data. The position of eye or the head plays an important role in calibration, thus it should be fixed throughout calibration as well as experiment. Subsequently, to achieve better controlling mouse pointer by eye, calibration is required every time. Accuracy of the tracker is directly proportional to successful calibration. It helps to determine eye movement properly, even in noisy environment, so that it can be used in practical domain. Different stages of calibration process are shown in Fig. 4. Figure 4(a) depicts the pupil detection process of *ITU GazeTracker*. From Fig. 4(b) and (c), we can see that how calibration is done and how accuracy varies in case of successful and unsuccessful calibration. If the accuracy is less than 5 stars, user experiences uneven movement of mouse and it becomes very hard to control. The reason of unsuccessful calibration is mostly due to involuntary head or eye movement and improper lighting conditions. For obtaining impressive result, we set the successful calibration threshold as 5 stars and less than 1° angle for *Monocular* eye detection.



(a) Pupil detection by ITU Gazetraker

(b) Successful calibration results and the points on the screen

(c) Unsuccessful calibration results and the points on the screen

Fig. 4. Different stages of calibrating eye

For participants, first few sessions are spent as training sessions, where they are briefed about the nature of the experiment and completed a short demographic questionnaire. They are introduced to the eye tracking hardware (camera and Infrared lamp positions) and the *EyeBoard* keyboard interface along with other designs (Fig. 5).

As discussed above, inability of users for moving the eyes beyond the visibility range of screen during text entry session is a major issue. So, as an way out, participants need to memorize the phrase to be typed before starting of experiments. In this context, participants first write the phrase

Fig. 5. Participant performing experiments

using pen and paper or listen while instructor prompting it. The total time needed for this interaction is about 10 minutes. After practicing on paper, participants are given two practice phrases with *EyeBoard* design as well as one of the other 3 designs, chosen randomly. First session lasts about an hour, and data are not considered for analysis. After completion of training, each participant on an average, composed 9 texts for testing. On average, each testing session takes about 45 minutes. For conducting experiments, 9 texts are selected and among these, 1 is taken from the in-domain Mayzner and Tresselts corpus and other 8 are taken from out-of-domain texts such as novels, short stories etc. for judging the efficacy of the designs.

H. *Hypothesis*

The hypothesis is that participants would take much time to learn the *EyeBoard* design and after learning the layout will outperform other designs in terms of text entry rate and errors committed in typing.

I. *Dependent measures*

The dependent measures used in this experiment are words per minute (WPM), uncorrected error rate, corrected error rate, and the total error rate [21], [22]. We kept dwell time for all designs, expect *EyeBoard*, as 700 ms.

1) Text entry rate: Let T be the final transcribed or typed string entered by the user and $|T|$ is the length of this string, that is, the number of characters entered. Let S denotes the time taken by a user in seconds, measured from the entry of the first character to the entry of the last, including backspaces. We define the text entry rate WPM [23] as shown in Eqn. 1. Here, $\overline{w}$ denotes average length of words for a language. It has been estimated that average word length of English is 5.

$$WPM = \frac{|T|-1}{S} \times \frac{60}{\overline{w}} \tag{1}$$

2) Corrected and Uncorrected errors: According to Gentner et al. [24], *Uncorrected* errors are the errors remaining in the transcribed string. On the other hand, *Corrected* indicates the errors which are already corrected in the text entry process [20].

III. EXPERIMENTAL RESULTS

Initial experiments are performed to calculate the layout area of 3 existing designs along with the *EyeBoard* design. Users are also involved for evaluating those designs with respect to *user friendliness* and *usability* perspectives. The results summarize that the *EyeBoard* system bears exact layout area which also satisfies the basic requirement of saccadic eye movement of users. The other designs are usually larger and not properly optimized with respect to expert users' eye movement during gaze typing. Apart from this, the main objectives of our experiments are to judge the following

1) Text entry rate
2) Corrected and uncorrected error rates
3) Learning curve

Data for each participant are averaged for each session to form single measures per participant per session on a variety of metrics, including entry rate in wpm and various error rates [21]. Participants complete a total of 4 trials × 4 designs × 9 sessions = 144 trials. With 8 participants, the entire study comprises 1152 trials.

A. *Text Entry Rate*

Longitudinal study with 4 designs, based on the average speed of different sessions, trials and results, is depicted in Fig. 6. It reveals that Design 2 yields 2.17% better text entry rate (4.23 wpm $SD = 1.07$) than Design 1. Similarly, Design 3 gives 4.35 wpm ($SD = 1.12$) which is 6.34% more than Design 1. On the other hand,*EyeBoard* achieves 20.83% more text entry rate than Design 1 (5.02 wpm and $SD = 1.46$). The analysis of variance on text entry speeds shows that there is a significant difference between the means of user's performance on different keyboard designs ($F(3, 1112) = 17.48$, $p < 0.05$). Also, for the sessions, significant difference are observed on wpm, as participants speed up with each design ($F(5, 278) = 7.22$, $p < 0.05$).

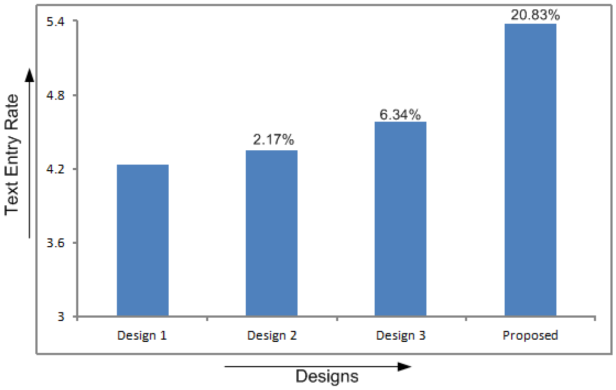


Fig. 6. Comparison among different designs

B. *Uncorrected Errors*

Uncorrected errors (Fig. 7) are errors left in the final transcribed text [21]. Thus, uncorrected errors are maintaining direct proportionality with speed, the more errors one commits, the faster one can go, and vice-versa. Over 9 sessions, the average uncorrected error rate for *EyeBoard* is 1.88%. On the other hand, for Design 1, 2 and 3 uncorrected error rates are 4.02%, 4.11% and 2.95%, respectively. The differences between designs which encourages less error among the existing and proposed design was significant ($F(3, 1112) = 3.83$, $p < 0.05$). The *EyeBoard* system achieved a mean of 1.066 KSPC in the experiments involving expert participants.

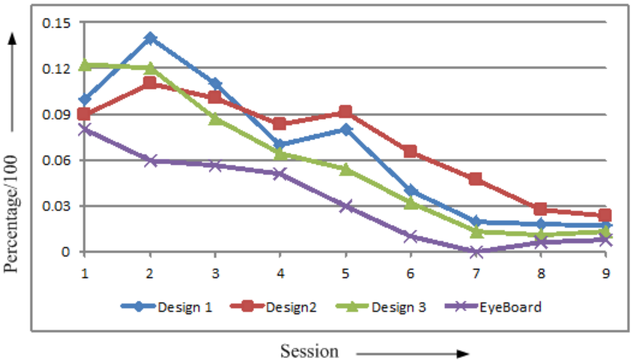


Fig. 7. Comparison between uncorrected errors of 4 designs

Looking at the graph of uncorrected errors over all sessions (Fig. 7), we see that in every cases, uncorrected error rates are higher for initial sessions, after that they become lower and stabilize at the end. For all sessions, *EyeBoard* achieves significantly less uncorrected errors left in the transcribed texts ($F(3, 1112) = 7.95$, $p < 0.05$). Experimental data also showed a significant effect of session on uncorrected errors ($F(5, 115) = 4.67$, $p < 0.05$).

C. Corrected Errors

Corrected errors (Fig 8) are those which are made and then corrected during text entry [21]. Thus, corrected errors reflect the scenario where a text entry method is error prone or not, it may produce accurate text in the end [22]. Over 9 sessions, the average corrected error rate for *EyeBoard* is 11.27%. On the other hand, for Design 1, 2 and 3, results are 9.82%, 9.77% and 10.95%, respectively. Thus, it seems all the interfaces exhibited approximately the same amount of error correction (i.e. backspaces) during entry.

Following a similar partitioning approach as for uncorrected errors, if we examine only the early sessions ($1-6$) in Fig. 8, we see a significant result in favor of *EyeBoard* ($F(3, 738) = 6.70$, $p < 0.05$). For the remaining sessions ($7-9$), the result becomes non-significant and poor (F(3, 372) = 1.80, n.s.). There is again a significant effect of session, as participants entered fewer backspaces over time ($F(3, 115) = 5.37$, $p < 0.05$).

D. Total Errors

Soukoreff and MacKenzie [21] define total error rate to be the sum of uncorrected and corrected errors. Over 9 sessions, the total error rate (Fig. 9) is 13.15% for *EyeBoard* and 13.84%, 13.88% and 13.90% for Design 1, 2 and 3, respectively. However, total error rates drop significantly over sessions ($F(3, 115) = 6.83$, $p < 0.05$).

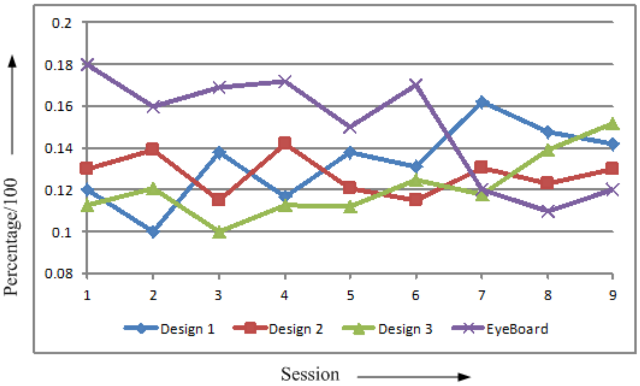


Fig. 8. Comparison between corrected errors of 4 designs

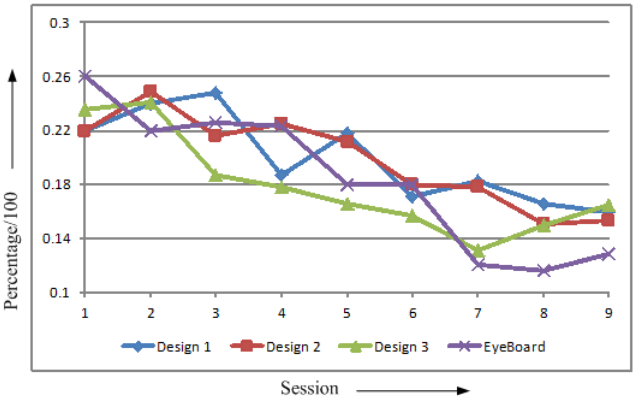


Fig. 9. Comparison between total errors of 4 designs

The results we got from the above error analysis do not strictly reflect better performance of the proposed *EyeBoard* system than other designs. In contrast, the observation reveals that using the proposed interface, users left a few errors uncorrected than other designs, i.e., the number of corrected errors is more in case of *EyeBoard* interface. Further, to get a clear picture, we analyze number of errors left in the transcribed text which is being indicated as accuracy measure, for all the 4 designs. An analysis of variance reveals that there is no significant difference in error rates between the keyboard designs ($F(3, 278) = 1.18$, n.s.).

E. Learning Curve

Learning curves are typically created for measuring task performance speed over time [25]. It does not reflect the study of initial interactions with the system, rather indicates that whether more or less training is required to get habituated. Figure 10 indicates that *EyeBoard* needs more initial effort to learn compared to Design 1, 2 and 3; however, after 20 to 25 sessions, *EyeBoard* outperforms other three designs. The *EyeBoard* layout reaches nearly 5.71 wpm by the 25^{th} session whereas the performance of the Design 1, 2 and 3 interfaces achieve up to 5.23 wpm text entry speed in average. We use standard *regression* models in the form of the curve fitting as it follows *Power law of learning*. The prediction equations and the *Squared correlation coefficients* for the curves are illustrated in Fig 10. The longitudinal study lasts for 60 sessions for each experienced and inexperienced users. We plot the results and construct the learning curves which inevitably reflect the increasing efficiency of users after performing several sessions (see Fig 10). So, the observation supports the previously stated hypothesis inevitably. The highest text entry rate achieved by user on typing similar length texts for the *EyeBoard* layout is 8.25 wpm and for the Design 1, 2 and 3, results are 7.31, 6.79 and 6.55 wpm, respectively.

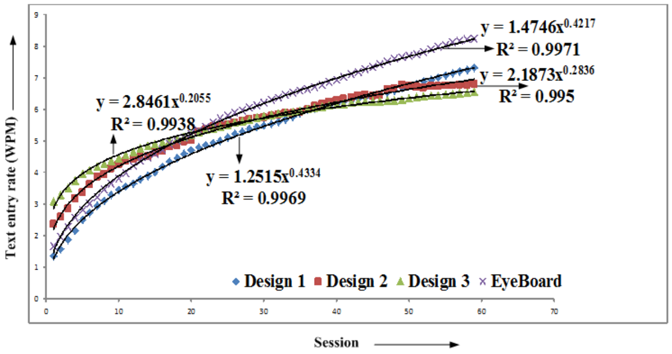


Fig. 10. Learning curve

F. Subjective Evaluation

We collect the subjective ratings from the participants with the *nonparametric Wilcoxon Matched Pairs Signed Ranks Test*. We talked with the participants before and after each session asking them about their eye strain and tiredness in a scale of 1 to 7. The level of tiredness is calculated by subtracting the first value from the later value. Analyzing the experimental results, we observe no significant difference between the average level of the tiredness, which is 0.52 in the first and 0.71 in the last session. We also calculate the text entry speed, ease of use, and general fatigue after each session using a questionnaire with a scale from 1 to 5. An increment of text entry rate is observed (3.4 to 4.6). On the other hand, the observed ease of use, with average rating of 4.5, and general fatigue (≈ 3.5) remain approximately on the same level. Finally, participants are again interviewed after completion of the series of sessions. Participants fell that typing by gaze is fairly easy, easier than their expectations, but clearly slower than using a conventional, hand operated hardware/virtual keyboard.

Participants think that they have improved in gaze typing over the sessions, especially in the beginning. All participants believe that typing speed adjustment is clear and easy to use. Dynamically dwell time adjustment is considered as more important, agreed by maximum participant. More or less, half of the participants experience problems in using the interface with vary short dwell times.

G. Discussion

By analyzing the experimental results, it is evident that *EyeBoard* design achieves faster as well as accurate text entry than the other designs. Nevertheless, more text entry does not achieve equivalent growth in accuracy. So, inevitably, a speed-accuracy trade-off is prominently present in current state-of-art. As the total error rate do not differ significantly between *EyeBoard* and other designs, we can say that participants have become equally proficient in correcting errors in all designs.

Also, on the basis of subjective evaluations performed by users, it has been decided that *EyeBoard* yields faster text entry rate than other designs. Also, the confined screen space in *EyeBoard* system offers an advantage over off-screen targets in limiting saccade distance to the dimensions of *EyeBoard*'s window. However, our results show that people can gaze type moderately fast and accurately using this simple, easy to learn keyboard design, supported with dynamically adjustable dwell time.

IV. CONCLUSION

There have been a number of gaze input applications in recent years as well as eye gaze is used in mobile environments. Due to inherent *Midas Touch* problem in gaze-based interfaces, dwell time is still the dominant command activation mechanism. In this scenario, the crucial factors affecting the speed-accuracy trade-off of gaze input are visual searching of the target and specifying dwell time conforming proper target selection. In this paper, we present a method to dynamically adjust dwell time according to user's comfort toward accessing the system.

Further, research can be carried out in many ways like minimizing visual search time of finding keys during gaze-based text composition, controlling mouse speed, implementing spell and grammar checker, diminishing the dwell time completely etc., which can improve text entry rate as well as accuracy of gaze-based text typing interfaces.

REFERENCES

[1] P. Majaranta and K. J. Räihä, "Twenty Years of Eye Typing: Systems and Design Issues," in *Proceedings of the Symposium on Eye Tracking Research & Applications*. ACM, 2002, pp. 15–22.

[2] P. Majaranta, A. Aula, and K. J. Räihä, "Effects of Feedback on Eye Typing with a Short Dwell Time," in *Proceedings of the 2004 symposium on Eye tracking research & applications*. ACM, 2004, pp. 139–146.

[3] D. Miniotas, O. Spakov, and G. Evreinov, "Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space," in *Proceedings of INTERACT*, 2003, pp. 137–143.

[4] M. H. Urbina and A. Huckauf, "Dwell Time Free Eye Typing Approaches," in *Proceedings of the 3rd Conference on Communication by Gaze Interaction*, 2007, pp. 3–4.

[5] H. Drewes, A. D. Luca, and A. Schmidt, "Eye-gaze Interaction for Mobile Phones," in *Proceedings of the Mobility Conference*. ACM, 2007, pp. 364–371.

[6] P. Majaranta and K. J. Räihä, *Text Entry Systems: Mobility, accessibility, universality*. San Francisco, CA: Eds. Morgan Kaufmann, 2007, ch. Text Entry by Gaze: Utilizing Eye-tracking, pp. 175–187.

[7] R. J. K. Jacob, "The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get," *ACM Transactions on Information Systems*, vol. 9, no. 2, pp. 152–169, 1991.

[8] H. Istance, R. Bates, A. Hyrskykari, and S. Vickers, “Snap Clutch, a Moded Approach to Solving the Midas Touch Problem,” in *Proceedings of the 2008 symposium on Eye tracking research & applications*. ACM, 2008, pp. 221–228.

[9] J. P. Hansen, D. W. Hansen, and A. S. Johansen, “Bringing Gaze-based Interaction back to Basics,” *Proceedings of Universal Access in Human-Computer Interaction*, pp. 325–328, 2001.

[10] J. M. Wolfe, “Guided Search 2.0 - A Revised Model of Visual Search,” *Psychonomic bulletin & review*, vol. 1, no. 2, pp. 202–238, 1994.

[11] O. Špakov and P. Majaranta, “Scrollable Keyboards for Eye Typing,” in *Proceedings of the 4th Annual Conference on Communication by Gaze Interaction*, Prague, Czech Republic, 2008, pp. 63–66.

[12] 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 Symposium on Eye Tracking Research & applications*. ACM, 2008, pp. 11–18.

[13] 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*. ACM, 2004, pp. 203–206.

[14] P. Majaranta, U. K. Ahola, and O. Špakov, “Fast Gaze Typing with an Adjustable Dwell Time,” in *Proceedings of the 27th international conference on Human factors in computing systems*. Boston, USA: ACM, 2009, pp. 357–360.

[15] 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*. ACM, 2012, pp. 241–244.

[16] S. Zhai, M. Hunter, and B. A. Smith, “Performance Optimization of Virtual Keyboards,” *Human Computer Interaction*, vol. 17, no. 2, pp. 229–269, 2002.

[17] A. Dix, J. Finley, G. Abowd, and R. Beale, *Human-computer Interaction*, 3rd ed. London, UK: Prentice-Hall Inc., 2004.

[18] M. S. Mayzner and M. E. Tresselt, “Tables of Single-letter and Digram Frequency Counts for Various Word-length and Letter-position Combinations,” *Psychonomic Monograph Supplements*, vol. 1, no. 2, pp. 13–32, 1965.

[19] J. F. Collins and L. K. Blackwell, “Effects of Eye Dominance and Retinal Distance on Binocular Rivalry,” *Perceptual Motor Skills*, vol. 39, pp. 747–754, 1974.

[20] R. W. Soukoreff and I. S. MacKenzie, “Recent Developments in Text Entry Error Rate Measurements,” in *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. Vienna, Austria: ACM, 2004, pp. 1425–1428.

[21] 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 conference on Human factors in computing systems*. ACM, 2003, pp. 113–120.

[22] J. O. Wobbrock and B. A. Myers, “Analyzing the input stream for character-level errors in unconstrained text entry evaluations,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 4, pp. 458–489, 2006.

[23] I. S. MacKenzie and K. Tanaka-Ishii, *Text Entry Systems: Mobility, Accessibility, Universality*. MA, USA: Morgan Kaufmann Inc., 2007.

[24] W. E. Kooper, *Cognitive Aspects of Skilled Typewriting*. Springer, 1983.

[25] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human Computer Interaction*. Hilldale, NJ: Elbaum Associates, 1983.