EyeK: An Efficient Dwell-Free Eye Gaze-Based Text Entry System

Sayan Sarcar
School of Information Technology
Indian Institute of Technology Kharagpur
mailtosayan@gmail.com

Prateek Panwar
Computer Science & Engineering
SDMCET Karnataka
prateekpanwar31@gmail.com

Tuhin Chakraborty
School of Information Technology
Indian Institute of Technology Kharagpur
tuhin.babai@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 minimizing dwell time, visual search time and interface area, 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 *EyeK*, a gaze-based text entry system which diminishes dwell time and favors to mitigate visual search time. Performance evaluation shows that proposed interface achieves on an average 15% higher text entry rate over the existing interfaces. As designed, the proposed interface can effortlessly be suited in medium-sized display devices like Tablet PC, PDA etc. Also, the developed system can be used by the people with motor disabilities.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces - Input devices and strategies

INTRODUCTION

Recently gaze control interaction has been firmly evolved as an alternate interaction method to support faster as well as accurate text entry in digital devices in last few decades [12]. This mechanism holds significant similarity with any traditional text entry methods; the only difference counted is that instead of hand, the controlling human organ is eye. Also, eye gaze-based text entry is among those few which, with the intact setup with able-bodied, can be extended toward applicability of disabled capable of visual interaction and having good vision. Many applications support text entry through eye gaze ([11, 15, 23]), even in the mobile environment [3].

Eye typing, the eye gaze-based text entry, is accomplished by direct pointing or looking at the desired letter within interface [13]. The computational requirements for *eye typing* include an on-screen keyboard and an eye tracking device. Key selection, i.e. *eye press*, is performed by hovering on the key for a slightly prolonged duration which is *dwell time*. Many gaze typing systems support *eye blink* besides eye movement during typing. Beside gaining popularity among alternate text entry mechanisms [9], gaze-based text entry mechanism posses a number of design issues which make it a unique technique with own set of research problems.

Number of keys and space between keys of the keyboard can be decreased to save screen space [15]. In contrast, bigger sized keys help user in easy typing even in a setup with low spatial resolution [5]. Consequently, in some cases, instead having fewer keys, the interface takes larger screen space. Thus, 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 visual stimuli presented in the interface such as color, orientation, shape, size, spatial frequency, etc. [26]. Optimizing screen area is one important constraint, specially for small display devices. Špakov and Miniotas [21] developed a keyboard that saves screen space as well as instantly usable without any special learning.

A typical issue in dwell-based eye pointing is 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 [6]. So, we cannot conclude that the shorter the dwell time, the better the 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 [13], stated that most gaze typing evaluations were conducted with novices using a constant, fairly long dwell time ($450 - 1000$ ms). Recently, Wobbrock et al. [25] used a short dwell time of 330 ms and achieved text entry rate of 7 wpm. Špakov and Miniotas [22], and Majaranta and Räihä [13] studied automatic adjustment of dwell time. Although the typing result of those systems were better, they pointed out *delay* and *involuntary variation* problems. Therefore, a trade-off still remains among dwell time, text entry rate and accuracy of the interface.

APCHI '13, September 24 - 27 2013, Bangalore, India
Copyright 2013 ACM 978-1-4503-2253-9/13/09 ... $15.00.
http://dx.doi.org/10.1145/2525194.2525288

Minimizing dwell time is another way to increase eye typing rate. Experimental analysis on eye typing reveals that throughout the experiment, dwell time may vary depending on user's comfort. Keeping this in mind, Majaranta et al. [10] developed an interface which dynamically adjusted dwell time during experiment. They got promising result in case of increasing eye typing rate. Urbina and Huckoff [23], Kristensson and Vertanen [7] proposed a research direction for eye-typing which is potentially much faster and dwell-free.

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

1. Proposing a new mechanism which diminishes the dwell time at the time of eye typing
2. Predicting and highlighting next probable characters after tapping a key to mitigate visual search time

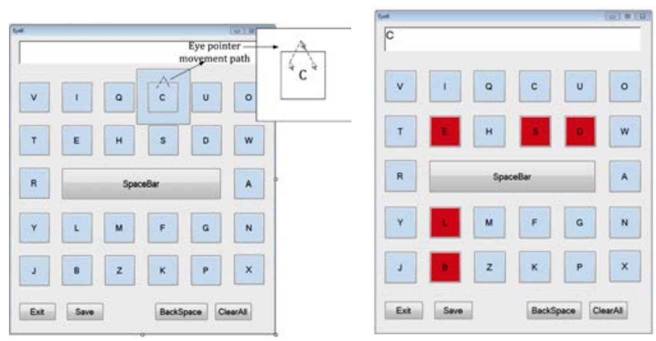
METHODOLOGY

We develop a gaze-based text entry system to cater the aforementioned issues. The proposed system is named as *EyeK*, which consists of a keyboard with *Eye typing* support following Panwar et al.'s *EyeBoard* layout [17]. It holds two design principles; placing 27 characters (space included) in a 5×6 matrix (almost square), and home row (3^{rd} row) holds a Space character of 4 key sizes (to increase reachability as it occurs much more than other characters in language texts: for English [14]). Further, other characters are placed into two concentric zones following the rule that higher frequent characters reside nearer to Spacebar followed by rest into other zone. The characters in each zone are placed in a row-major order in accordance with their frequencies in descending order. Then, depending on the bi-gram frequencies between each character pair, characters are rearranged following *Trial and error* method. In addition, two critical constraints regarding eye movement and fixation behavior need to be analyzed in the proposed *Eye Typing* interface; a) eyes always tend to move, not to fix around a point for moderately large amount of time and b) sudden change in color contrast among visual contents in an interface always draws attention [18].

Mechanism supporting Dwell-free Eye Typing

Previously, Kriestensson and Vertanen [7] proposed an unique eye-gaze based text entry mechanism which supports eye typing without spending dwell time while selecting each character. The proposed system recognizes the sequences of characters traveled by users eye gaze in the interface and identifies the valid combination of words from dictionary. Since the system eliminates dwell-timeouts for key selections, it becomes potentially quicker than state-of-the-art dwell based eye-typing interfaces. The major concern about this interface is, for people who are not well conversant of the layout, it is not of worth accessing. Moreover, the detailed design methodology of the Kriestensson and Vertanen's work has not been released to the community.

As dwell time indicates the time eye fixes on the object, our dwell free interaction is based on objects also. In the eye typing keyboard, every key area is divided into two parts, inside actual key area and outside overlay area. Both height and width of the actual keys are 1.6 cm. We checked the length of overlay area starting from 0.4 cm with an increment of 0.2 cm and found 0.8 cm as the comfortable length of overlay area where all of our participants performed the character selection task very smoothly. So, we keep the length of overlay area from the boundary of actual key area as 0.8 cm in our design. The interaction actually activates while user moves the eye pointer through the areas in inside-outside-inside fashion, shown in Fig. 1(a). With this oriented movement, keys will be selected automatically. As eye movement is faster than mouse or finger movement, this interaction takes minimum effort and time which is negligible with respect to dwell time. A pictorial example clarifies the scenario further. Suppose, user wants to select character 'C'. While hovering on the character, key and outside area are visible (if it is highlighted previously, then after hovering, highlighting will be off). User starts moving the eye pointer from the key area, goes to outside area and again comes back to key area (almost following a trajectory) to complete the interaction phase (Fig. 1(a), enlarged portion). After coming back on the key, character is entered (Fig. 1(b)). If users need to enter same characters twice, they have to get out of the character initially and enter it again in similar manner.



(a) Eye movement on a key (b) The key gets selected

Figure 1. Dwell time free eye typing in *EyeK* interface

Minimizing visual search time

User feedback over a long period of time on different keyboard designs reveals that, for novice as well as experts, visual search task takes significant amount of time for character based text composition through gaze. Unlike to mouse or touch based interface, eye movement time includes visual search time for a gaze-based typing interface. From the analysis of Sears et al. [19], it has been clearly summarized that visual search time does not vary only with number of keys present on keyboard interface, rather depends on certain other features like size of the keys, distance between keys, different color of the key groups etc. Few papers in the domain of mouse or touch-based interfaces pointed out the importance of visual search time and tried to mitigate it in mobile devices [8, 4]. They only concentrated on size of the keys and font size of the characters present in the keyboards. We design many alternate layouts to reduce the time as well as make them tested with the users. Finally, we come up with a design which considers size of the key, character font size and boldness features of the keys. The character level bigram and trigram values are calculated from Mayzner and Tresselt's table [14]. On the other hand, based on the last two consecutive characters typed, the system automatically predicts next characters. Effectively, at any instance, the five most likely next characters are highlighted in the keyboard (the number is not rigid as in any instance, if number of next probable characters becomes less than five, then remaining candidates are displayed). As an example, after composing character chunk 'SC', *EyeK* chooses next probable characters as 'E', 'A', 'O', 'H' and 'K' and colors them with red (Fig. 2). This scenario increases user comfort as well as diminishes error committing tendency of frequent keyboard users. Although the keyboard occupies larger area on the screen, this method helps common users to confine their eye movements within the specific region and selects the intended character from highlighted keys, in most of the cases.

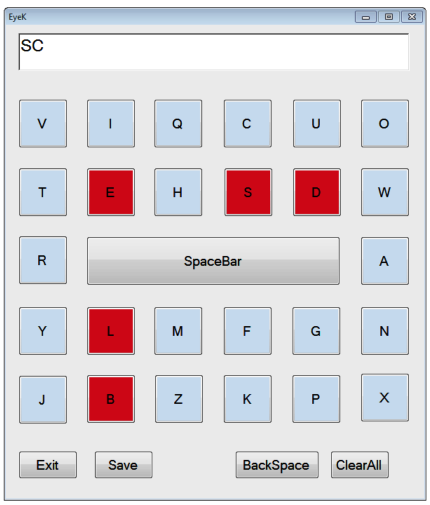


Figure 2. Next character highlighting in *EyeK* interface

EXPERIMENTAL SETUP

Following the aforementioned approaches, we have designed *EyeK* eye-typing interface. Before conducting user experiments, we have prepared setup which are described below.

Apparatus

All experiments were conducted in a low cost eye tracking setup using 2.2GHz Intel Core2Duo processor with $15''$ wide screen LCD color monitor having 1440×900 resolution. Modified *Sony PlayStation Eye* webcam, original lens was 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 [1], developed by IT University of Copenhagen, were used for experiments. The key press events and gaze positions were recorded automatically and stores into log files using separate event hooking programs. All experiments were performed in Windows 7 environment. Controlled light conditions and positioning of the setup were maintained.

Participants

Five participants (3 male, 2 female) were recruited from the local university campus. Participants ranged from 20 to 31 years (mean = 22.5). All were regular computer users, access on an average 4 hours per day, but no prior experience with eye tracking. All participants except one had normal vision and expertise in composing text through digital devices. 4 participants were right-eye dominant and 1 are left-eye dominant, as determined using an eye dominance test [2].

Designs

Six designs including *EyeK* were chosen for experiments. The designs are namely, optimized *scrollable* keyboard which saves the screen space proposed by Špakov et al. [21], keyboard designed by Majaranta et al. [10] maintaining adjustable dwell time, a design called *Iwrite*, which is a square shaped interface keeping characters at outer side and text area in middle for gaze typing [23] (Design 3), Morimoto and Amir propose *Context Switching* (CS) as a new activation mechanism for gaze controlled interfaces [16] and Panwar et al.'s key size and space optimized *EyeBoard* [17] layout with adjustable dwell time. In the experiments, 9 texts were considered for typing. Each text contains 10 phrases each 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[14]. The result is $r = 0.973$ for the single-letter correlation and $r = 0.908$ for the digraph correlation. Each participant performs 9 sessions, each for a corpus, for each of the 6 keyboards.

Procedure

To perform user-based evaluation, users first synchronized their eye movement with the gaze tracker through *Calibration* followed by typing session where mouse pointer was moved with eye gaze. Inability of users in moving their eyes beyond the visibility range of screen during the session was a major issue of the experiment. Alternatively, participants first wrote the phrase using pen and paper or listened while instructor prompting it. The main objective was to compose the phrases as fast as possible committing few errors. Correcting errors was possible by erasing text using backspace and then retyping it.

Before the experiments, participants spent first few sessions for training where they were briefed about the nature of the experiment and completed a short demographic questionnaire. They also got familiarized with eye tracking hardware (camera and Infrared lamp positions) and the *EyeK* keyboard interface along with other designs (Fig. 3). The total time for this interaction was about 10 minutes.

After practicing on paper, participants were given two practice phrases with 6 designs, appeared at random order. The order was counterbalanced across the participants. First session took about one and half hour, and data were not considered for analysis. After completion of training, each participant on an average, composed 9 texts for testing. Before starting of each sessions, users assured the instructors about their memorability of the practiced set. On average, each testing session took about 45 minutes. 9 texts were selected for experiments and among these, 1 was taken from the in-domain Mayzner and Tresselts corpus and other 8 were taken from out-of-domain texts such as novels, stories etc. for judging the design efficacy.

Hypothesis

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

Dependent measures

The dependent measures used in this experiment were words per minute (WPM) and the total error rate [20, 24].

EXPERIMENTAL RESULTS

Within subject experiments were performed to calculate the layout area of 5 existing designs along with the *EyeK*interface. Users were also involved for subjective evaluations with respect to *user friendliness*, *usability* etc. The results summarized that the *EyeK* system maintains compact layout area supporting saccadic eye movements 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 were to judge a) Text entry rate and b) Total error rate. Data for each participant were averaged for each session to form single measures per participant per session on a variety of metrics, including entry rate in wpm and error rates [20]. Participants completed a total of 4 trials $\times$ 6 designs $\times$ 9 sessions $= 216$ trials. With 5 participants, the entire study comprised of 1080 trials.

Text Entry Rate

Result of user experiments with 6 designs, based on the average speed of different sessions, trials and results, is depicted in Fig. 4. It reveals that *Adjustable dwell time*-based design yields 2.17% better text entry rate (4.23 wpm $SD = 1.07$) than *Scrollable keyboard*. Similarly, *Context switching* and *Iwrite* supported eye-typing interface also give 4.35 wpm ($SD = 1.12$) and 5.05 wpm ($SD = 1.18$) text entry rates

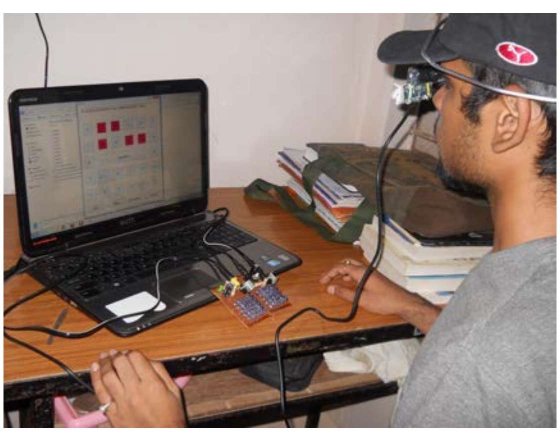


Figure 3. Participant performing experiments

which are 6.34% and 14.86% more than *Scrollable keyboard* interface. *EyeBoard* gets 5.25 wpm ($SD = 1.05$) entry speed which is 18.48% better. *EyeK* achieves 32.61% more text entry rate than *Scrollable keyboard* interface (6.03 wpm and $SD = 1.16$). The analysis of variance (ANOVA) on text entry speeds shows that there is a significant difference between the means of user's performance on different keyboard designs ($F(5, 260) = 6.42$, $p < 0.05$). Further, The Post-hoc using *Tukey HSD* test reveals significant difference between performance of *EyeK* and other keyboard designs ($p < 0.05$). Also, for the sessions, significant difference is observed on wpm, as participants speed up with each design ($F(8, 260) = 4.23$, $p < 0.05$).

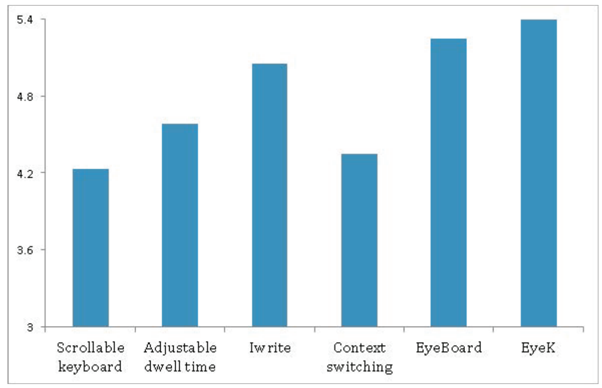


Figure 4. Comparison among different designs

Total Errors

Over 9 sessions, the total error rate, on an average, (Fig. 5) is 16.90% for *EyeK* and 19.90%, 19.82%, 17.90%, 18.64% and 19.36% for *Scrollable keyboard*, *Adjustable dwell time*-based design, *Iwrite*, *Context switching*-based interface and *EyeBoard*, respectively. However, total error rates drop significantly over sessions ($F(8, 260) = 4.29$, $p < 0.05$).

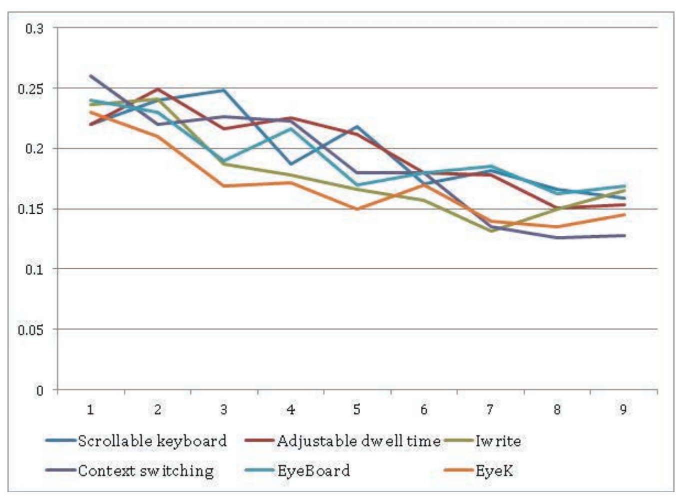


Figure 5. Comparison between total errors of 6 designs

The results we got from the above error analysis do not strictly reflect better performance of the proposed *EyeK* system than other designs. In contrast, the observation reveals that using the proposed interface, users left less errors uncorrected than other designs, i.e., the number of corrected errors is more in case of *EyeK* interface. We also analyze number of errors left in the transcribed text for all the 6 designs. An analysis of variance reveals that there is no significant difference in error rates between the keyboard designs ($F(5, 260) = 1.21$, n.s.).

Subjective Evaluation

We collected 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 1 to 7 Likert-scale. The result reveals that users liked the dwell-free eye-typing interface than other keyboard designs for ease of use ($z = 57.00, p < .001$) and less stressfulness ($z = -55.00, p < .001$). They found the *EyeK* interface more faster ($z = 51.00, p < .01$) and thus fun to use. However, users agreed that concentration was needed for eye typing through proposed keyboard, but they could improve their eye typing skill with practice easily. They also felt that gaze typing was clearly slower than using a conventional hardware keyboard.

Discussion

By analyzing the experimental results, it is evident that after training, users are able to achieve faster text entry using *EyeK* than other popular designs. Nevertheless, speed-accuracy trade-off is prominently present in current state-of-art. As the total error rates are not differed significantly between *EyeK* and other designs, we can say that participants have become equally proficient in correcting errors in all designs. The computer proficient people, at least, are not now wasting time by searching the key as *EyeK* suggests them by highlighting which attracts human eye. This scenario proves to be moderately fast while user is comfortable with the interface. On the other hand, this interaction after introducing dwell-freeness is initially seems to be harder to users. But, as the learning rate is quick, users pick up the speed after few eye typing sessions and feeling comfortable afterwords. The limited screen space acquired in *EyeK* system also offers an advantage over off-screen targets by limiting the saccade distance within the small dimensions of *EyeK*'s window.

Threats to validity

Relating to our experimental setup, experimental procedure and experimental results, we would like to point out their validity and limitations.

- Kriestensson and Vertanen had developed a dwell-free eye typing interface [7] with increasing the eye typing rate. The paper described a novel work in the related field but authors did not provide the working methodology of the system. So, further, this method could not be replicated which, in turn, hinders the possibility of comparing this with other systems.
- Currently, we have developed a low-cost eye tracking set-up which can be easily replicated. However, the accuracy of this still is not up to the mark and thus, the applicability confines within performing experiments in controlled environments. We further have tried to improvise the situation by fixing the infrared (IR) filters within visible range and placing the camera as close to eye for more accurately detecting eye gaze during calibration phase.
- We performed user experiment with 6 keyboards and 6 participants where each participant, in training session, practiced with 2 texts per keyboard. In case of avoiding redundant text occurrence, we required to collect at least $6 \times 6 \times 2 = 72$ texts whereas, practically, we gathered 8 texts each having 10 phrases (a text consists of several phrases; total $8 \times 10 = 80$ phrases). In a situation, if user is interested to type more than 2 texts for testing or number of participants gets increased, in current scenario, we are bound to encourage redundancy. On the other hand, sample text pool needs to be updated regularly which, in this work, was not handled.
- The developed concept *Next character highlight* presented in this paper is in a very premature stage where many cases are not handled. Currently, the module does not work for predicting the first letter of a word, after tapping space character. The proposed character prediction methodology works on simple character level bi-gram or tri-gram language models. The method can not predict the next character(s) properly when error occurs within the typed character chunk. Presently, number of characters to be highlighted through the interface is fixed (which is 5). It should be varied depending on context which is not implemented in present system. Also, if number of next probable characters is less than 5, then all candidates are displayed in the interface.
- In the subjective evaluation section, the paper lacks in providing the questions those were asked to participants. This problem occurs because we usually asked a common question to all participants as "How do you feel after typing through 54 existing keyboards and proposed *EyeK* interface?". As the answers are usually large and general rather specific to any point, we summarize them in generic way.

CONCLUSION

There have been a number of gaze input applications in recent years even 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 that diminishes the dwell time concept and minimizes visual search time in finding keys. Overall, user evaluation, both based on text entry rate and subjective parameters, ensures that the newly proposed *EyeK* gaze based text entry system, which is unique in its kind, is acceptable to the people and can be evolved as a strong alternate to existing text entry mechanisms.

The analysis on user error rate requires many user experiment data which we are lacking of. So, presently we are collecting data which can further lead us to a decision which we could not achieve in this work. Further, research can be carried out in many ways like controlling mouse speed, implementing spell and grammar checker etc., which can improve text entry rate as well as accuracy of gaze-based text typing interfaces.

ACKMOWLEDGEMENT

We express our sincere thanks to IIT Kharagpur Research Scholar Mr. Pradipta Kumar Saha for their significant contributions in building the low cost experimental setup, respectively.

REFERENCES

1. Agustin, J. S., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., and Hansen, J. P. Evaluation of a Low-Cost Open-Source Gaze Tracker. In *Proc of ETRA*, ACM (New York, NY, USA, 2010), 77–80.
2. Collins, J. F., and Blackwell, L. K. Effects of Eye Dominance and Retinal Distance on Binocular Rivalry. *Perceptual Motor Skills 39* (1974), 747–754.
3. Drewes, H., Luca, A. D., and Schmidt, A. Eye-gaze Interaction for Mobile Phones. In *Proc of the Mobility Conference*, ACM (2007), 364–371.
4. Gong, J., Haggerty, B., and Tarasewich, P. An Enhanced Multitap Text Entry Method with Predictive Next-letter Highlighting. In *Extended Abstracts on Human Factors in Computing Systems*, ACM (New York, NY, USA, 2005), 1399–1402.
5. Hansen, J. P., Hansen, D. W., and Johansen, A. S. Bringing Gaze-based Interaction back to Basics. *Proceedings of Universal Access in Human-Computer Interaction* (2001), 325–328.
6. Jacob, R. J. K. The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get. *ACM Transactions on Information Systems 9*, 2 (1991), 152–169.
7. Kristensson, P. O., and Vertanen, K. The Potential of Dwell-free Eye-typing for Fast Assistive Gaze Communication. In *Proc of ETRA*, ACM (2012), 241–244.
8. Magnien, L., Bouraoui, J., and Vigouroux, N. Mobile Text Input with Soft Keyboards: Optimization by Means of Visual Clues. In *Proc of Mobile Human-Computer Interaction*, ACM (New York, NY, USA, 2004), 197–218.
9. Majaranta, P. *Text Entry by Eye Gaze*. PhD thesis, Department of Computer Science, 2009.
10. Majaranta, P., Ahola, U. K., and Špakov, O. Fast Gaze Typing with an Adjustable Dwell Time. In *Proc of conference on Human factors in computing systems*, ACM (Boston, USA, 2009), 357–360.
11. Majaranta, P., Aula, A., and Räihä, K. J. Effects of Feedback on Eye Typing with a Short Dwell Time. In *Proc of ETRA*, ACM (2004), 139–146.
12. Majaranta, P., and Räihä, K. J. Twenty Years of Eye Typing: Systems and Design Issues. In *Proc of ETRA*, ACM (2002), 15–22.
13. Majaranta, P., and Räihä, K. J. *Text Entry Systems: Mobility, accessibility, universality*. Eds. Morgan Kaufmann, San Francisco, CA, 2007, ch. Text Entry by Gaze: Utilizing Eye-tracking, 175–187.
14. Mayzner, M. S., and Tresselt, M. E. Tables of Single-letter and Digram Frequency Counts for Various Word-length and Letter-position Combinations. *Psychonomic Monograph Supplements 1*, 2 (1965), 13–32.
15. Miniotas, D., Spakov, O., and Evreinov, G. Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In *Proceedings of INTERACT* (2003), 137–143.
16. Morimoto, C. H., and Amir, A. Context Switching for Fast Key Selection in Text Entry Applications. In *Proc of ETRA*, ETRA '10, ACM (New York, NY, USA, 2010), 271–274.
17. Panwar, P., Sarcar, S., and Samanta, D. EyeBoard: A Fast and Accurate Eye Gaze-based Text Entry System. In *Proc of IHCI*, IEEE (2012), 1–8.
18. Pomplun, M., Reingold, E. M., and Shen, J. Area Activation: A computational Model of Saccadic Selectivity in Visual Search. *Cognitive Science 27*, 2 (2003), 299–312.
19. Sears, S., Jacko, J. A., Chu, J. Y. M., and Moro, F. The role of visual search in the design of effective soft keyboards. *Behaviour & Information Technology 20*, 3 (2001), 159–166.
20. Soukoreff, R. W., and MacKenzie, I. S. Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In *Proc of the conference on Human factors in computing systems*, ACM (2003), 113–120.
21. Špakov, O., and Majaranta, P. Scrollable Keyboards for Eye Typing. In *Proc of COGAIN* (Prague, Czech Republic, 2008), 63–66.
22. Špakov, O., and Miniotas, D. On-line Adjustment of Dwell Time for Target Selection by Gaze. In *Proc of NordiCHI*, ACM (2004), 203–206.
23. Urbina, M. H., and Huckauf, A. Dwell Time Free Eye Typing Approaches. In *Proc of COGAIN* (2007), 3–4.
24. Wobbrock, J. O., and Myers, B. A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction (TOCHI) 13*, 4 (2006), 458–489.
25. Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proc of ETRA*, ACM (2008), 11–18.
26. Wolfe, J. M. Guided Search 2.0 - A Revised Model of Visual Search. *Psychonomic bulletin & review 1*, 2 (1994), 202–238.