GazeTheKey: Interactive Keys to Integrate Word Predictions for Gaze-based Text Entry

Korok Sengupta
Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau
Koblenz, Germany
koroksengupta@uni-koblenz.de

Raphael Menges
Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau
Koblenz, Germany
raphaelmenges@uni-koblenz.de

Chandan Kumar
Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau
Koblenz, Germany
kumar@uni-koblenz.de

Steffen Staab
Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau
Koblenz, Germany
Web and Internet Science Research Group, University of Southampton, UK
staab@uni-koblenz.de

IUI'17 Companion, March 13-16, 2017, Limassol, Cyprus.
ACM 978-1-4503-4893-5/17/03.
DOI: http://dx.doi.org/10.1145/3030024.3038259

Abstract
In the conventional keyboard interfaces for eye typing, the functionalities of the virtual keys are static, i.e., user's gaze at a particular key simply translates the associated letter as user's input. In this work we argue the keys to be more dynamic and embed intelligent predictions to support gaze-based text entry. In this regard, we demonstrate a novel "GazeTheKey" interface where a key not only signifies the input character, but also predict the relevant words that could be selected by user's gaze utilizing a two-step dwell time.

Author Keywords
Gaze input; eye tracking; text entry; eye typing; dwell time; visual feedback.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces

Introduction
Gaze-based text entry is a valuable mechanism for the people with severe motor disabilities, who have no or little control of muscles to operate the physical keyboard. Usually, eye typing is accomplished by onscreen keyboard, where the user selects the letters by looking at them for certain duration called dwell time [4]. Researchers have

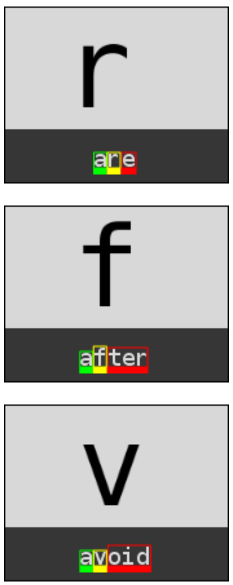

Figure 1: Suggestion per key is estimated with previous letters entered by user, and letter associated with the key as input. In the image above the already collected part of the word is framed green, the letter on the key as yellow, and the suggestive part by prediction engine red.

proposed alternate designs like Dasher [8], or dwell-free Filteryedping [7] to achieve high performance, which needs extensive training and expertise for faster and accurate input. Therefore, to assist novice users, most of the commercial and popular gaze-based systems[1] follow the convectional design of QUERTY key layout, associated with text area and word predictions. In the context, word predictions becomes a significant aspect, since it can reduce the number of keystrokes required to write the word and decrease user's effort while typing. However, for eye typing there is an additional cost of perceptual and cognitive load, caused by shifting the focus from the keyboard layout to the suggested word list and the repeated scanning of the list. Moreover, users are overloaded by the task of typing and might overlook the suggestions from an external area as they are engaged in the visual search task of finding the next key to enter the intended letter. Hence, there is a need to bring the predicted words within the visual search area of users. Furthermore, in the conventional design the word list is typically limited to three or four words, due to the space limitation of eye controlled interfaces [2], e.g., large buttons to tackle eye tracking accuracy.

In this work, we move beyond the conventional design of text suggestions as few predicted words on the top of keyboard layout, and propose a dynamic approach of embedding the suggestions on the key, which would not only bring the relevant suggestions in the visual attention of users, it would also offer the possibility to include much more suggestions.

GazeTheKey

The two major component of GazeTheKey (GTK) design is: how does the keys represent the word suggestions, and how does the user interact with these suggestions. The representation basically relies on the next word that the respective language model could predict based on the previously entered text by user [9]. Figure 1 shows the representation sketch of the keys containing the letter and the associated word suggestion at the bottom. The green framed letters on the key are the ones that have been already entered by user. This is succeeded by the letter on the key, framed in yellow. The red frame signifies the suggestive letters predicted based on previous letters.

While Figure 1 presents the structure of word suggestion on keys, Figure 2 displays the user interaction methodology via different states of a key on eye gaze fixation. **a)** At first, the key is in its standard mode to trigger the input of the single letter displayed on it. **b)** This is implemented via a dwell time that is visualized through an orange circle. When the user's gaze is upon the key, the circle grows from the center of the key until the key's area is completely filled. **c)** The input of the displayed letter is triggered and a visual feedback in appearance of a black pulse as feedback to the user. **d)** If a suggestion is available by the prediction engine, the key now turns into suggestion mode **e)** Further gaze fixation by the user lets the area of the suggestion fill the key starting from bottom and ending at the top. **f)** When the key is filled after this second dwell time, the currently collected word is replaced by the given suggestion. All key actions can be aborted by the user by looking at a different position on the screen.

Figure 3 shows the complete design of GTK keyboard interface including above-mentioned functionality (detailed demonstration of GTK usage is available here[2]). The design includes the principles of eye-controlled interfaces [2] (e.g., enlarged buttons and visual feedback to cope with eye

[1] https://github.com/OptiKey/OptiKey/wiki

[2] https://www.youtube.com/watch?v=-UDDTJHBPVA

(a) Standard mode.

(b) First dwell time.

(c) Letter input.

(d) Suggestion mode.

(e) Second dwell time.

(f) Suggestion input.

Figure 2: Key interactions.

tracking accuracy), moreover, it follows the usability heuristics of Neilsen [6], keeping the design as close to conventional keyboard layout as possible with minor adjustment of key formation and addition of necessary keys for improving efficiency while typing.

In the shown example in Figure 3, user typed some letter of a word and the current state of all keys shows relevant suggestions that can be selected via additional dwell time over the key (as explained in Figure 2). One limitation of such a two-step dwell time input is that the user wouldn't be able to dwell on a letter consecutively to type the letter multiple times. Hence an extra *repeat key* has been added at the lower bottom of the environment. This key serves the purpose to trigger the input of the last selected letter. We have done some further optimization to minimize the visual search for users, i.e., space and backspace key is used to present further information about the currently edited word. As shown in the example, the space button works as the confirmation of the typed word and therefore it is displayed on the bottom of key. A preview of the edited word (after deletion by backspace) is displayed on the backspace key (essentially showing the usage of respected key). These simple heuristics help the user to stay with the gaze on the same position without the need to check the intended action on the edited word.

Implementation

The proposed keyboard has been developed as experimental eyeGUI [5] element in C++. eyeGUI is a graphical user interface framework for eye tracking driven applications. Rendering is handled by OpenGL function calls and the FreeType2[3] library is used for font rasterization. For the relevant word predictions, Presage[4] library has been employed. The dictionary of the prediction engine is composed of a random subset of 50.000 English sentences from the Tatoeba[5] database. The proposed design presents several word predictions as one suggestion per key is to be computed. Therefore, an asynchronous calculation has been set up to guarantee an interruption-free interaction, otherwise the rendering may freeze after letter input for some seconds and user's gaze would have no effect on the interface.

[3] http://www.freetype.org

[4] http://presage.sourceforge.net

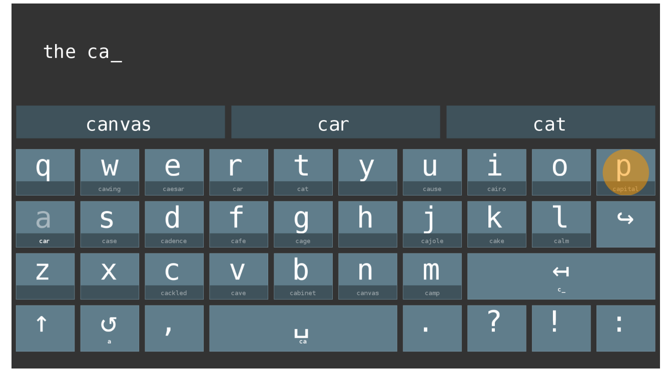

Figure 3: The new keyboard with characters predicting the next possible word if character were chosen. The repeat key on the left bottom row repeats the previously typed in letter.

Evaluation

We have conducted an initial small-scale study with *SMI REDn* remote eye tracker to investigate how the proposed design is able to support the task of eye typing. Ten participants (5 male, 5 female) contributed in our study, they were aged between 21 and 30 years (mean 24.8, SD 2.347) and had no prior experience with eye controlled interfaces. We asked the participants to type 25 different sentences (in five

[5] https://tatoeba.org/eng

sessions), which were randomly chosen from the phrase set by MacKenzie and Soukoreff [3]. Participants found the design intuitive, and often gazed at keys to pick recommended suggestions. The results indicates that 54.4% of all the word suggestions were selected via gazing on the keys. Highest of 92.6% was reached by one of the participants in a session. Moreover, despite of its novelty to end-users, GTK recorded text entry rate of 9.34 words per minute (max = 11.17 wpm) showing a significant acceptability. Longer training period should further enhance GazeTheKey performance.

Conclusions

We proposed GazeTheKey interface to bring the relevant suggestions in the visual attention of users, minimizing the additional cost of scanning an external word suggestion list. Furthermore, it offers the possibility to include much more suggestions than the conventional interfaces having few suggestions at the top of keyboard layout. In future we plan to conduct extensive evaluation of the new design against the other state of the art text entry systems. Moreover, since the new design aims to lower the perceptual and cognitive load of users, we plan to quantify the mental workload (with EEG sensors) of the new design in comparison to conventional word prediction interfaces.

Acknowledgments

This work is part of project MAMEM [1] that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement number: 644780.

REFERENCES

1. 2015-2018. Multimedia Authoring and Management Using Your Eyes and Mind. `http://www.mamem.eu/`. (2015-2018).
2. C. Kumar, R. Menges, and S. Staab. 2016. Eye-Controlled Interfaces for Multimedia Interaction. *IEEE MultiMedia* 23, 4 (Oct 2016), 6–13.
3. I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. ACM, 754–755.
4. Päivi Majaranta. 2009. *Text entry by eye gaze*. Tampereen yliopisto.
5. Raphael Menges, Chandan Kumar, Korok Sengupta, and Steffen Staab. 2016. eyeGUI: A Novel Framework for Eye-Controlled User Interfaces. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 121, 6 pages.
6. Jakob Nielsen. 1995. 10 Usability Heuristics for User Interface Design. https://www.nngroup.com/articles/ten-usability-heuristics/. (1995).
7. Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N Truong. 2015. Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing (TACCESS)* 6, 1 (2015), 3.
8. Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash away!: longitudinal study of fast text entry by Eye Gaze. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. ACM, 19–26.
9. Matteo Vescovi. 2004. Soothsayer: un sistema multi-sorgente per la predizione del testo. (2004).