Text Editing by Gaze: Static vs. Dynamic Menus

Päivi Majaranta[1], Niina Majaranta[1], Gintautas Daunys[2]. and Oleg Špakov[1]

[1] Department of Computer Sciences/TAUCHI
33014 University of Tampere, Finland
{forename.surname}@cs.uta.fi

[2] Siauliai University
Vilniaus str. 141-413, Siauliai, Lithuania
g.daunys@cr.su.lt

Keywords
Gaze typing, text entry, gaze input

Introduction

Eye tracking enables communication and computer control by gaze for people with severe disabilities: the gaze direction is used to point at items on a computer screen, for example, to point at a letter on an on-screen keyboard (Majaranta & Räihä, 2007). Dwell time (a prolonged gaze longer than the time needed for normal viewing, typically >500 ms), is used to distinguish the intentional command from casual viewing.

Even though it is easy to point at items by gaze, direct mapping of the eye movements into "eye mouse" coordinates is problematic. The accuracy of the measured point of gaze is usually about 0.5 degrees, after a successful calibration. However, due to drifting, the practical accuracy is often much worse (from 1 degree up to several degrees). This means that selection of small items is hard by gaze alone, without supporting techniques such as zooming (Bates & Istance, 2002; Skovsgaard et al., 2008).

There are several practical implications for the inaccuracy. Since the measured point of gaze has drifted a few pixels off from the actual focus point, it is difficult to place the cursor exactly on the desired location in the body text. Furthermore, a gaze operated on-screen keyboard also requires large buttons that may allocate most of the available screen space. Since the keyboard itself takes a lot of space, there is not much space left for editing commands (such as copy, paste, bold, italics, etc.). Gaze based text entry systems typically provide a backspace or undo key for immediate corrections. However, the editing commands are often hidden in the virtual keyboard's menu structure.

Using the same modality for both input and output presents another kind of challenge for the interface design. Since the gaze is needed for selecting the button, the user cannot see the effect on the text simultaneously but needs to leave the keyboard area to review the result of the action on the text.

We developed a dynamic pie-like menu that can potentially facilitate the task of text editing by gaze. A pie menu is a pop-up menu that appears at the place of focus. The menu items are placed in a circular pattern around the center of the pie. Pie menus have been proved to be faster than ordinary linear menus in normal mouse-based interaction (Kurtenbach & Buxton, 1994). Our assumption is that by having the editing commands near the focus of interest can facilitate the editing process by gaze. In this paper, we first briefly review related research. We then introduce our prototype of a dynamic pie menu for text editing by gaze and report preliminary results from the first user trial in which we compared the dynamic pie menu with a static editing menu. Finally, we discuss ideas for further improvements and future work.

Related Research

Huckauf and Urbina (2007) developed a pie menu based system "pEYEs" (also called pEYEwrite) for gaze based text entry. The letters were grouped into the sectors of the pie menu. To type a letter, the user first selects the sector that contains the desired letter. It opens a sub-menu that contains a separate sector for each of the letters in the group. No dwell time is needed, since the focused sector is selected by looking at or crossing a special selection area on the outer edge of each sector. The letters are located so that the most common letter of each group is located at the direction of the sector, thus the user can select a letter in a long continuous gaze gesture into that direction. For example, a look up selects the sector containing 'e', and another look into the same direction (up) selects the subsector 'e' in the sub-menu.

Tien and Atkins (2008) tested different menu layouts for gaze interaction: a layout that resembled a typical drop-down menu, a layout resembling typical gesture based menus found in hand-held devices, and a variation of the gesture based menu adjusted for gaze, with big buttons and near distance. They did not find significant differences in task times between the layouts. After the initial experiment they implemented several improvements for the menu designed specifically for gaze, for example, a "snap-on" feature that fixed the eye mouse cursor to the centre of the button, and a feature that opened the menu with a quick off-screen glance to the left. In the follow-up experiment they found that after memorizing the menu commands, participants were able to perform menu selections using dwell times as short as 150 or 180 ms.

Also Kammerer et al. (2008) experimented with three different designs of multi-level menus operated by gaze. They found that a semi-circle menu was better suited for selection by gaze than a full-circle layout or a linear (conventional) menu design. Based on participants' subjective experience, Kammerer et al. stated that the major drawback of the full-circle menu was its confusing arrangement (widespread and ungrouped menu items) and long distances between menu items.

Dynamic Pie Menu for Text Editing by Gaze

We implemented a prototype of a gaze operated dynamic pie menu for text editing (illustrated in Figure 1 on the left). The pie menu is shown on the point of the user's focus when the user fixates on the text for longer than the predefined dwell time (1500 ms). The cursor (caret) is located in the center of the pie. The user can see the text through the central hole and also the menu items are partially transparent.

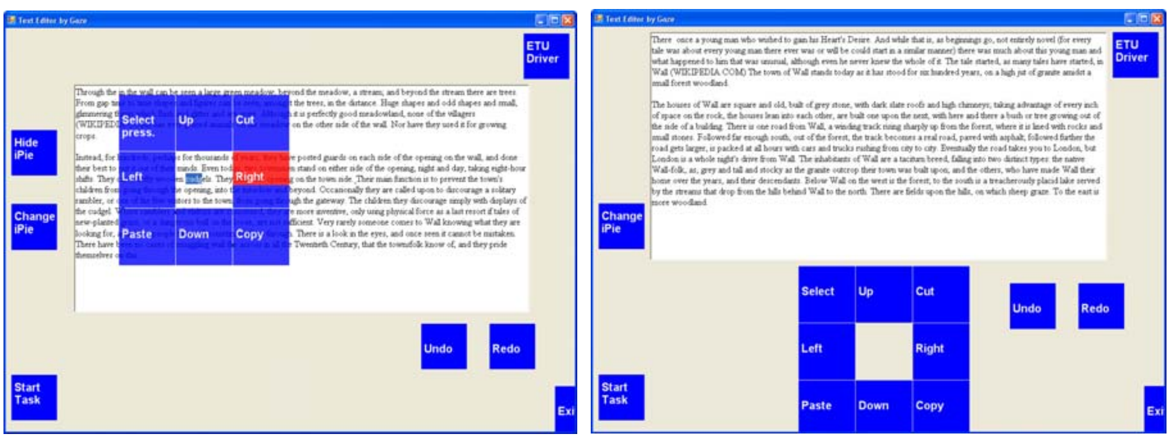

Figure 1. Dynamic pie menu (on the left) and static menu (on the right). The image on the left also illustrates feedback shown on the progression of the dwell time on the 'right' menu key.

The user can fine-tune the (often misplaced) cursor position using the left, right, up, and down keys. The pie menu moves along so that the cursor is always located in the center. To select text, the user needs to press (dwell on) the 'Select' button, and then move the cursor using the direction keys. An editing command such as 'Copy' can then be executed for the selected text. Our experimental prototype included functions for basic text editing (cut, copy, paste) and text formatting (bold, italic, underline). The commands on the pie menu can be changed by using the 'Change iPie' button (located on the left side of the application window).

In addition to the dynamic pie menu, we also implemented a static menu that stays at a fixed location on the bottom of the screen (see Figure 1, on the right). The layout and functionality of the keys were the same for both menus. The dwell time for selecting a button was set to 1000 ms for both. If the user kept on looking at the button it started to repeat the click with an interval of 450 ms. Durations are based on pilot tests.

Method

We conducted an experiment with 13 participants (10 males, 3 females, 19-26 years, mean 21) to study the usability of the dynamic pie menu for text editing by gaze. Participants were university students, with good computer skills and average to good text editing skills. All were novices in editing text by gaze but two had some previous experience on gaze control and one had participated in an eye tracking related experiment. The Tobii 1750 eye tracker was used with the COGAIN ETU Driver to track the gaze.

The experiment was a within subjects study with two conditions: dynamic pie menu and static menu (illustrated in Figure 1). The participants were assigned into two groups; participants in the first group started with the dynamic pie menu, and participants in the second group started with the static menu.

The participants were first briefed about gaze interaction and the experiment. They were informed that we wanted to compare two different interfaces for editing text by gaze. They then filled in a pre-questionnaire. Each test started with calibration. Before the actual test, the experimental software was introduced to the participants, starting with the condition that was assigned to the participant, and they had a change to practice using it with two simple tasks and to ask any questions.

During the test, each participant finished six similar tasks with both interfaces. Each task started with the press of the 'Start Task' button and ended by selecting 'End Task'. Participants started with four simple formatting tasks, for example, to select a word and bold it. The last two tasks were text editing tasks, where the participant had to move a word or swap the locations of two words using the cut and paste commands.

After finishing all six tasks with one condition, the participants were interviewed about the first design. The same procedure was then repeated with the second condition, starting with the introduction of the interface and practice, and ending with the interview (the same questions were asked from each user after both conditions). After finishing both conditions, the participants filled in a questionnaire where they had a chance to compare the two designs and we interviewed them.

Preliminary Results

We lost data from several tasks from several participants due to technical problems. A few participants had poor calibration, which affected their performance. In addition, there was a bug in the experimental software that we noticed only after the tests had begun. Therefore, we will not report statistically significant results for the performance measurements; instead, in this paper we focus on reporting initial user reactions and ideas for further improvement.

Despite loosing some of the data, we did look into task completion times (including only successfully finished tasks with no bugs, compared with the corresponding tasks from the other condition for each participant). There seems to be a trend in the task times indicating that the participants performed faster in the simple formatting tasks (tasks 1-4) using the dynamic pie menu (with the average grand total of 38 seconds) compared to the static menu condition (aver. total 47 sec). However, when completing the more complex editing tasks (tasks 5-6), they performed slower using the dynamic pie menu (with the average grand total of 77 seconds) compared to the static menu (aver. total 67 sec).

Out of the 13 participants 8 preferred the static menu over the dynamic pie menu (preferred by 5) if they had to choose only one. If they had a chance to use both, 5 would still prefer using the static design only, 3 would prefer dynamic and 5 would like to use both (especially after further practice, as noted by a couple of participants).

We were interested in the usefulness of having the navigation keys on the dynamic pie menu, since we assumed they would be especially useful for adjusting the location of the cursor in the text, even if all other functions (formatting and editing) were placed on a static menu. We asked the participants if they felt it was easier to use the arrows on the dynamic or the static menu. Five participants felt it would be best to have the arrows on the static menu (to avoid confusion, as some of them commented). Others felt placing the arrows on the dynamic menu was indeed a good idea (4 participants), or probably a good idea (4 participants).

We also asked the participants to rate which one of the designs was faster, easier to use, more comfortable and easier to the eyes. The dynamic and static design both got equal number of votes for being faster (6/6, plus 1 "cannot say" reply). The static design got more votes for all other categories (number of votes for static/dynamic/cannot say): easier (9/3/1), more comfortable (7/5/1), easier to the eyes (7/3/3).

During the interview, we asked what was most difficult in using each of the menus. For the dynamic pie menu, 3 participants felt that the menu disturbed visibility (of the text under it or the interface in general), and 3 felt it was difficult to swap the gaze between the menu buttons and the text (to see the effect of e.g. selection). For the static menu, 4 participants felt that switching between the menu and the text was difficult, and 3 participants felt placing the cursor on the correct place was difficult. Other difficulties observed by more than one participant were related to gaze interaction in general, such as the difficulty of fixating in the same location for long enough, or a feeling of rush when the dwell time was running out, or difficulties related to the implementation of the experimental software. For example, we had implemented a feature that automatically hid the dynamic menu if the user looked at the grey area around the text field for longer than the threshold. However, there was a bug in the implementation since it sometimes caused a disappearance of the dynamic menu in the middle of text editing.

For both designs, a few participants complained that it was hard to select a full word and to remember how many times they had to eye-press the "left" or "right" to select all of the letters in the word. Some participants suggested that there should be an option to select a full word, or an option to define the starting and ending point for the selection (instead of repetitive presses of the left or right arrow). Even though most participants felt that the static menu was easier to use and perceive as it stayed in the familiar location, they also wished it was nearer to the text. Several participants wished they could adjust the transparency level of the dynamic menu's buttons. Other suggestions for improvement for the dynamic menu included replacing the buttons' caption texts with icons, having more options (more buttons or sectors for the pie) or placing the 'Change iPie' button (which swaps between the formatting and editing menus) into the dynamic menu itself for easy access. Some also felt the buttons were unnecessarily big and too far apart.

Discussion and Future Work

Even though most participants preferred the static menu over the dynamic pie menu, we believe there is potential in having the editing commands in a dynamic pie menu. First, we observed a trend in having faster task competition times using the dynamic pie menu for simple formatting tasks. Second, some participants preferred the dynamic pie menu and several more would like to have both options available. It is worth noting that there were more bugs in the dynamic pie menu condition than in the static menu condition, which may have affected the participants' subjective experience (even though we asked them to ignore the bugs in their ratings). The current experiment was very short and participants were novices in gaze interaction, therefore we believe the difficulties related to gaze interaction in general may have also affected the results. More practice would be needed to see the full potential of both solutions. Thus, we plan to organize a longitudinal experiment after improving the design (and correcting the bugs, obviously).

We agree with the participants that the text in the menu buttons should be replaced with icons. Icons would be fast to recognize and would not disturb the visibility of the text as much as the current design which has a partially transparent text buttons over the body text.

In the current implementation, the buttons in the dynamic pie were basically normal dwell time activated buttons. In the future design, we want to test a menu that looks like and operates more like a pie menu: it would be circular with sectors near each other. The sectors could also be selected by simply looking over the outer edge of the sector (similarly as in the pie menu design by Huckauf and Urbina, 2007): as long as the user views the command icon in the sector, nothing would happen but as soon as the gaze crosses the sector's outer edge, the sector would be selected – or a new sub-menu (with sub-sectors) would be opened. For example, the basic layout could have the arrows (as icons) and other sectors for formatting and editing. Those could open a next level of command, for example, activating the formatting sector could show a sub-menu for bold, italics and underline.

Our prototype did not allow using the dynamic pie menu near the edge of the screen and that is why a fairly large empty (grey) area was added around the text field. This problem could be solved by implementing the half-circle layout suggested by Kammerer et al. (2008). Dynamically changing the orientation of the half- (or partial) circle layout could easily compensate for the lack of space in one direction.

To our knowledge, editing text by gaze has not been researched. This pilot study is the first step towards more user friendly text editing by gaze. We believe this area offers a rich set of opportunities for future research and development.

References

Bates, R. & Istance, H.O. (2002) Zooming interfaces! Enhancing the performance of eye controlled pointing devices. *Proceedings ASSETS '02*, ACM Press, 119-126.

Huckauf, A. & Urbina, M. (2007) Gazing with pEYE: new concepts in eye typing. *Proceedings APGV '07*, ACM Press, 141-141.

Kammerer, Y., Scheiter, K., & Beinhauer, W. 2008. Looking my way through the menu: the impact of menu design and multimodal input on gaze-based menu selection. *Proceedings of ETRA '08*, ACM Press, 213-220.

Kurtenbach, G. & Buxton, W. (1994) User learning and performance with marking menus. *Proceedings of CHI'94*, ACM Press, 258-264.

Majaranta, P. & Räihä, K.-J. (2007) Text Entry by Gaze: Utilizing Eye-Tracking. In I.S. MacKenzie & K. Tanaka-Ishii (Eds.), *Text entry systems: Mobility, accessibility, universality*. San Francisco: Morgan Kaufmann, 175-187.

Skovsgaard, H.H.T., Hansen, J.P., & Mateo, J.C. (2008) How can tiny buttons be hit using gaze only? *Proceedings of COGAIN 2008*. CTU Publishing House (ISBN 978-80-01-04151-2), 38-42.

Tien, G. & Atkins, M. S. (2008) Improving hands-free menu selection using eyegaze glances and fixations. *Proceedings of ETRA '08*. ACM Press, 47-50.