Meta-keys: Extending the Functionality of Gaze-based Interaction

Antonio Diaz Tula
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
Rua do Matão 1010, São Paulo, Brazil
diaztula@ime.usp.br

Carlos H. Morimoto
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
Rua do Matão 1010, São Paulo, Brazil
hitoshi@ime.usp.br

ABSTRACT

This paper introduces the concept of meta-keys with the objective to extend the currently limited functionality of gaze-based interaction. Meta-keys are two-step gaze gestures between the interface and external markers. A bridge between the interface and the markers avoids accidental activation and reduces the effect of the eye tracking noise. Results of a user study showed that meta-keys have the potential to extend the functionality of gaze-based interaction with a low error rate.

Author Keywords

gaze-based interaction, meta-keys, gaze gesture

ACM Classification Keywords

H.5.2 Information Systems: INFORMATION INTERFACES AND PRESENTATION; H.1.2 Information Systems: MODELS AND PRINCIPLES

INTRODUCTION

Gaze-based interaction represents an alternative of communication for people with disabilities, such as Amyotrophic lateral sclerosis and Locked-in syndrome. This kind of interaction uses as input an eye tracker, a device that informs the position of the user's gaze on a computer monitor.

To effectively communicate with a computer in this modality of interaction, the user must be able to point and select objects that are presented on the monitor using his or her eyes. One of the most common tasks using gaze is text entry ("eye typing"), since it allows people with disabilities to communicate with the world, express their feelings and improve their quality of life. Although several interfaces for eye typing have been proposed and investigated, once the text is typed, it must be sent by email to someone, saved into a document or redirected to another application, for example. Therefore, escape mechanisms are needed to change the focus between different applications using the gaze.

Because of the inaccuracy and noise in gaze estimation of current eye trackers, graphical elements in gaze-based interfaces are bigger than in other interfaces that use more accurate pointing devices, like mouse or joysticks. This restriction limits the screen space avaliable to present graphical elements in gaze-based interfaces. For example, in a virtual keyboard controlled by gaze, the keys should be big enough to avoid wrong selections. This implies that a virtual keyboard cannot show both characters and digits at the same time. Another example of limited screen space is when the user is browsing a collection of objects that cannot be shown in the interface at the same time. Therefore, there should be a mean to switch between different layouts or views of the interface.

In this paper we extend the work of Tula et al. [16] by introducing the concept of "meta-keys" to improve the usability of gaze-based interaction. Meta-keys allow the user to execute additional commands easily and with a low error rate, like changing the layout or browsing different views of the interface. To show the use of meta-keys in practice, we report the results of an experiment where participants had to navigate a collection of alphanumeric characters and select all digits using the their gaze as fast as possible. Results showed evidence that meta-keys have the potential to complement the functionality of gaze-based interaction with a low rate of false activation.

GAZE-BASED INTERACTION TECHNIQUES

The most common technique for gaze interaction is *dwell time*. In this paradigm of interaction, a virtual keyboard is shown on the screen and the user must fixate on the desired key for a period of time to select it, usually between 600 and 1000 ms [10]. Examples of interfaces based on dwell time are ERICA [8] and GazeTalk [3].

Jacob [7] recognized a potential problem of selection by gaze called "midas Touch". This problem refers to the involuntary selection of any observed object. The fixation time required to select keys in dwell-based virtual keyboards is a way to reduce the Midas touch problem. Nonetheless, a non-trivial design issue is to choose the most appropriate fixation time. Shorter fixation times improve performance since less time is needed to select a key. However it also increases the error rate since the time available to correct a fixation over a wrong key is very short. Novice users, unfamiliar with the keyboard

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. IHC'14, Brazilian Symposium on Human Factors in Computing Systems. October 27-31, 2014, Foz do Iguaçu, PR, Brazil. Copyright 2014 SBC. ISSN 2316-5138 (pendrive). ISBN 978-85-7669-291-1 (online).

layout, can commit a lot of errors with short dwell times. On the other hand, longer dwell times are less error prone, but the interaction becomes slower and can cause eye fatigue.

The use of discrete gaze gestures for gaze interaction have emerged as an alternative to dwell time. According to Heikkilä and Räihä [4], gaze gestures are patterns of eye movement used to issue commands. An example of interface that uses gaze gestures for eye typing is EyeWrite [19], where each character is mapped to a sequence of eye movement between the corners and the center of a 400 × 400 window. A similar interface was presented by Porta and Turina [14] called Eye-S, but different from EyeWrite, it uses nine points to execute the gestures. Another example is the MDITIM interface from Isokoski [6] that uses off-screen areas around the screen (Nort, South, East and West) as hotspots. Characters are mapped to the sequence of tokens produced when the user look at these off-screen areas.

Discrete gaze gesture techniques need less screen space than virtual keyboards and do not suffer the midas Touch problem (as long as the gestures can be robustly differentiated from regular exploration of the interface). However, they are more difficult to learn since users must memorize the mapping of the gestures to the characters (at least 26 gestures are needed to type the entire english alphabet). Discrete gaze gestures also exhibit a lower performance than virtual keyboards, since several saccades are needed to complete a gesture. Furthermore, prolonged use can cause eye fatigue.

Other techniques interaction techniques are based on continuous gaze gestures. Perhaps the most cited example is Dasher, from Ward and MacKay [18]. In Dasher, the user must follow the desired character with the gaze as its moves from the right side of the screen to the center, while zooming in the area around it. Selection is completed when the character crosses the center of the screen. Dasher needs a lot of screen space since all characters are arranged in a single column. Its performance is better than other techniques because it uses a model language to predict the next characters to be entered. Another example of interface based on continuous gaze gestures is StarGazer, by Hansen et al. [2]. This interface uses pan and zoom in the direction of the gaze to simulate navigation in a 3D world until the desired character enters the selection area located in the center of the screen. It can be used with small screens and is robust to noise conditions, although its performance is lower compared to Dasher and to virtual keyboards.

Continuous gaze gestures techniques have the drawback that the user is always controlling the interface. Any change in the direction of gaze could be interpreted by the interface as a command from the user, therefore there is no suitable place to rest the eyes.

Some techniques use a combination of both fixations and saccades to interact using the eye gaze. Pie Menu [5, 17] is an interface based on hierarchical circular pie menus designed for text entry. In a Pie Menu, characters are grouped into circular sectors. To select a character, the user must fixate on the corresponding sector for a short dwell time or cross the sector border (from inside the sector to the area outside the circle) to expand a second pie menu containing all the characters from the selected sector. Following a similar procedure, the user can select the sector with the desired character from the second circular menu. Experiments with Pie Menus have shown that after several sessions, users "learned" the path sequence to select characters by making a continuous gaze movement. Since Pie Menus need a lot of screen space, only one line of typed text is shown to the user.

Q	W	E	R	T	Y	U	I	O
A	S	D	F	G	H	J	K	L
Z	X	C	V	SPC	B	N	M	P

Aperte a barra de espaco para comecar.

Q	W	E	R	T	Y	U	I	O
A	S	D	F	G	H	J	K	L
Z	X	C	V	SPC	B	N	M	P

Figure 1. Virtual keyboard for eye typing based in the Context Switching paradigm, image reproduced from Morimoto and Amir.

THE USE OF META-KEYS TO EXTEND THE INTERFACE FUNCTIONALITY

It can be seen from the literature review that most gaze-based interaction techniques focus on pointing and selection of elements, like characters. We propose meta-keys as an extension of gaze-based interfaces to execute additional functionality, like changing the layout of the interface, browsing several views and execute additional commands.

To provide a good usability to the users, meta-keys should be efficient and easy to execute, learn and remember. In order to have a good user experience, they should also minimize the number of accidental activation.

General purpose commands using single gaze gestures

The use of gaze gestures for general purpose commands has already been proposed by Porta and Turina [14]. If the gestures are too comples, like in EyeWrite [19] and Eye-S [14], they could be difficult to learn and cause eye fatigue [14]. On the other hand, if the gestures were too simple (only one saccade), like the "single stroke gaze gesture" suggested by Møllenbach et al. [11], they could be activated while the user is exploring the interface.

Drewes and Schmidt proposed the use of gaze gestures that can be executed anywhere [1], hence needing neither an accurate calibration of the eye tracker nor an specific screen area to execute the gestures. Nonetheless, this may result in accidental command activation while the user is exploring the visual content. The use of specific areas to make the gesture helps the user to have more control over the gesture execution, and at the same time provides visual feedback. To save screen space, the gesture could be executed using off-screen targets, as proposed by Isokoski [6].

To define the meta-keys, we decided to use two-step gaze gestures with visual markers to guide the gesture sequence. Next we briefly describe this interaction paradigm, highlighting the main features that were used to complement the definition of meta-keys.

The Context Switching paradigm

Context Switching [13] is a paradigm for gaze-based interaction that uses only one saccade per selection. Objects (keys) are grouped within areas called contexts, and contexts are separated by a "bridge". The user can freely explore a context without risk of unintentional selection, thus avoiding the Midas touch problem. When the user looks at an object for a short time (typically 150 ms) the object receives the focus. The selection of the key in focus is made by saccading to the other context, crossing the bridge entirely. The bridge avoids switching contexts accidentally and also reduces the effect of the eye tracker noise. The bridge can also be used to show additional information to the user, like the typed text. Different from the dwell time paradigm, Context Switching clearly separates focus and selection, associating focus to eye fixations and selection to saccades. As a result, users can to naturally adjust their selection speed without the need of adjusting any other parameter. Figure 1 shows a virtual keyboard based on this paradigm.

One limitation of the Context Switching paradigm is the use of considerable space in the monitor, because two contexts are visible all the time. Tula et al. [16] proposed the use of contexts with dynamic size to overcome this limitation. In their work, the authors used two-step gaze gestures to navigate between several pages of alphanumeric characters. In this paper, we extend the authors' work by formally defining the meta-keys and discussing the results together with the dynamic contexts.

Location of the markers in the interface

To define the location of the markers to execute the meta-keys, we were inspired by the idea of Morimoto and Amir [13], that to execute an action the user must cross the bridge between two contexts.

Markers are located around the interface, with a bridge between the interface's edges and the marker. The distance between the interface and the marker, i.e., the bridge width, must be short to make the gesture comfortable, but at the same time, it should be long enough to avoid unintentional activation and reduce the effect of the eye tracker noise. Therefore, this distance can be expressed as a function of the eye tracker precision. To activate a meta-key, the user can start from anywhere inside the interface, then make a saccade to the marker, and finally saccade back to the interface, crossing the bridge two times. This sequence must be completed within a maximum time interval to guarantee that it is indeed a gaze gesture.

Figure 2 shows an implementation of the meta-keys concept. The marker (small blue square) is located close to the left edge of the context (3 × 2 grid of light-blue keys), between the first and the second rows and displaced to the left. Note that a bridge separates the meta-key from the context, acting as a filter for the eye tracker noise. To activate the meta-key in the example of Figure 2, the user must begin within the context, in a key located in the first or the second (as in the figure) row of the context, then look at the marker, and finally look back to the context again, to a key in a row adjacent to the one where the gesture began. The requirement of making the gesture between two adjacent rows is specific for the implementation presented in Figure 2 and therefore is not part of the concept of meta-keys. Because the interface in the example have a grid layout, we decided to include this restriction to reduce the chances of accidental activation. Note that in Figure 2 the marker is located alongside the vertical edge of the contexts, but it can be alongside the horizontal edge as well.

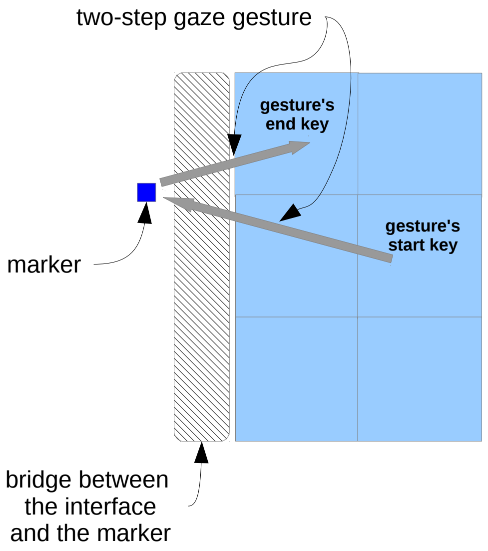


Figure 2. Implementation of the meta-key concept. The gesture to activate the meta-key is formed by two saccades, one from a key to the marker, and the other from the marker to a key in a row adjacent to the one where the gesture began. The bridge must be crossed two times.

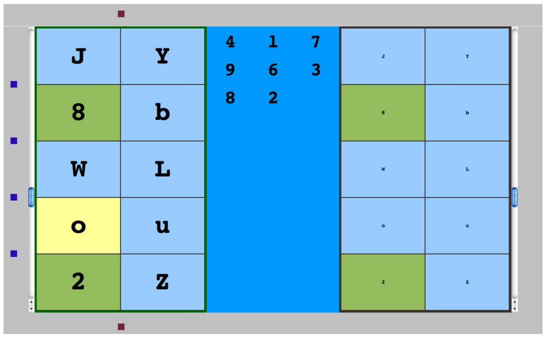
In the next section we describe a user experiment that was conducted to evaluate the use of meta-keys together with the Context Switching paradigm.

METHOD

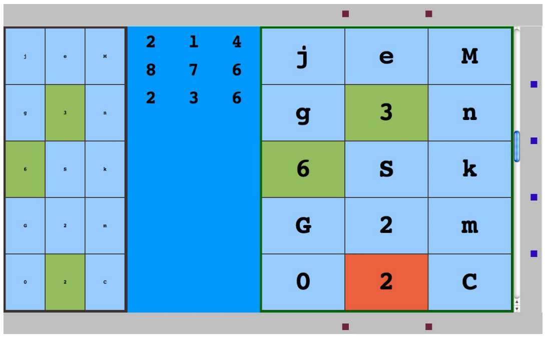
To evaluate the use of meta-keys, we have designed a user experiment consisting of a multiple selection task, which required browsing through several pages of items. This is a very common task in real world applications, such as navigating a collection of pictures or multimedia objects.

Participants

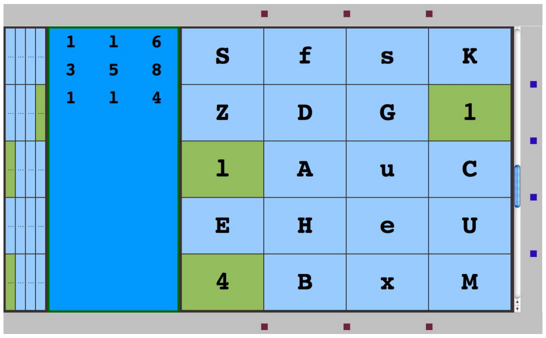
A total of 6 people participated in our experiment. They were all male, able-bodied, with normal or corrected to normal vision. Two participants had never used an eye tracker before, two had already participated in at least one study involving eye trackers for gaze interaction, and the other two had



a) Two columns layout



b) Three columns layout



c) Four columns layout

Figure 3. Three different layouts used in the experiment.

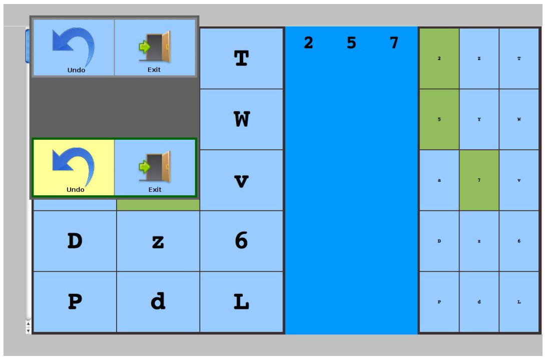


Figure 4. Menu activated with a meta-key along the vertical edge of the contexts.

experience developing eye trackers and participating in gaze interaction studies. Because it was a pilot study, we did not recruit participants from the group of final users, i.e., people with disabilities.

Apparatus

A low-cost, pupil-corneal reflexion eye tracker described by Morimoto et al. [12] was used during the experiment. The eye tracker runs at 30 Hz and has about 1^o in visual angle of spatial accuracy.

Figure 3 shows the three different layouts that were developed for the experiment. The layouts had two columns (2C), shown in Figure 3a, three columns (3C), as can be seen in Figure 3b, and four columns (4C), shown in Figure 3c. All layouts had five rows. In all configurations, markers (squares near the edges of the contexts) were placed between columns and rows of the contexts, indicating the locations to activate the meta-keys.

In the 2C configuration the size of the contexts was kept constant. In the 3C and 4C, the context with the user's focus was displayed in full size, while the other context was minimized. The size of the keys, as well the bridge between the two contexts, were kept constant in all configurations. The bridge between the two contexts was used to present the selected items. A short dwell time of 150 ms was used for detecting focus on a virtual key, and the maximum time for selection by context switching (i.e. maximum duration for the saccade) was set to 450 ms for the 2C and 3C configurations, and to 550 ms in the 4C configuration (because on average this layout requires longer saccades to switch contexts).

Meta-keys along the vertical edges of the contexts were used to bring up a menu with options to undo the last selection and start/finish each session. A example of a menu is shown in Figure 4. Meta-keys along the horizontal edges of the contexts (located above and below the contexts) were used to switch pages. To move to the previous page, for example, starting from any column, the user can look at a top marker (associated to page-up) and then look at an adjacent column (left or right) within the same context. Similar gestures can be used to activate meta-keys along the vertical edges of the contexts, starting from any row. To provide proper visual feedback, markers changed their color when the user looked at them.

Experimental design

The study was a within subjects design, where participants used all the three conditions. The order of the conditions were varied according to a latin-square design.

The task itself was to selected digits (numeric characters) from a collection of alphanumeric characters (lower and upper case letters from the English alphabet). This reduced the cognitive load of participants during the experiment, so they could focus on the interaction. The total number of alphanumeric characters per trial was fixed to 120 for all configurations. Since we wanted to make participants browse through all pages using the meta-keys, the number of digits to be selected in each trial was picked randomly within the interval [18, 28], uniformly distributed. This correspond to about 15% to 25% of the total number of alphanumeric characters in the collection. If a fixed number of digits were used, a participant could count the number of selections and exit the trial as soon as the count reaches the fixed number of digits.

Before beginning the experiment, participants signed a consent form and were introduced to the study. After the introduction, all participants performed a training session that lasted about 10 minutes. In the training session, participants used all the three different conditions in random order, with the objective to learn the task and operate the gaze interface. Each session, including the training session, started with the calibration of the eye tracker. During calibration participants had to look at nine different points in the screen and press the space bar for each one. The calibration process was repeated until a reasonable precision was obtained, as evaluated by the experimenter.

After the training session, all volunteers participated in 6 sessions that last about 15 minutes each. In each session the participant had to perform 9 trials, 3 for each layout. In each trial, the user was told to select the digits as fast as possible, and to be careful not to leave digits unselected. A session could not be repeated within 30 minutes, so that most volunteers took 2 or 3 days to complete their sessions.

If a participant lost calibration during a session, results of that trial were discarded and the user repeated the trial. There where cases when a trial could not be repeated because of the participant's schedule, so fewer trials were considered for those participants in the data analysis. At the end of the experiment participants were interviewed and answered a questionary. Both the interview and questionary were designed to collect participants' impressions of the interaction using the three layouts and the meta-keys.

Data Analysis

To estimate the time needed for the execution of meta-keys, we separate the time users dedicated to selection from the time spent for paging. Let's VP be the set of visited pages during a trial, and TS the number of final selected items in the task (considering all pages).

For every visited page in VP, the *selection time* is defined from the moment the page was shown to the last selection done within that page. Because each page could have a different number of selections, we also computed the *Selection time per digit* (STPD) for each page, by dividing the selection time by the number of selections in that page. The *Average selection time* (AST), i.e., the time needed to make a single selection (independently of the paging time) for each configuration is computed as follows:

$$AST = \frac{1}{|\text{VP}|} \sum_{p \in \text{ VP}} STPD_p \tag{1}$$

The *paging time* is defined for each page from the last selection within that page to the execution of a meta-key to go to the next (or previous) page. The *Average paging time* (APT), i.e., the mean of the paging time for all visited pages for each configuration, is computed as follows:

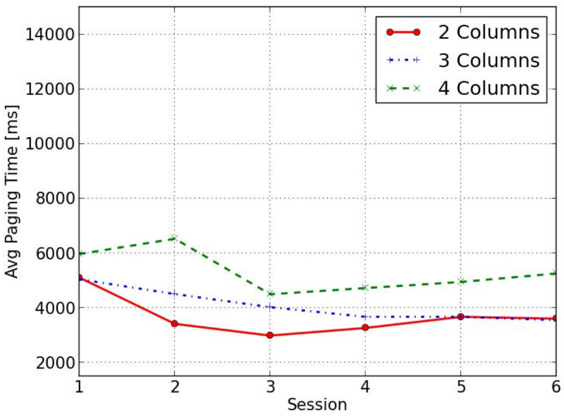


Figure 5. Grand mean for Average paging time (APT).

$$APT = \frac{1}{|\text{VP}|} \sum_{p \in \text{ VP}} \text{paging time}_p \tag{2}$$

Finally, to evaluate the time needed to complete the task, including the selection and paging time, we defined the *Average time task* (ATT) for each configuration as:

$$ATT = \frac{\text{total time task}}{TS} \tag{3}$$

EXPERIMENTAL RESULTS

The grand mean of Average paging time for the six participants is shown in Figure 5. As can be observed, the 4C layout had longer APT than the 2C and 3C layouts. Mauchly's test did not show a violation of sphericity distribution of APT values (W=0.77, p=0.59). A one-way repeated measures ANOVA found a significant main effect of layout on APT (F(2,10)=22.1, $p < 0.05$). A post-hoc test with Bonferroni correction showed that APT in the 4C layout was significantly longer that in 2C and 3C ($p < 0.05$ in both cases). There was not significant difference in APT between 2C and 3C.

Figure 6 shows the grand mean for the Average selection time, computed from the six participants for the three layouts. It can be observed that the 4C layout had a much higher AST than the other two layouts, while the 2C layout had the shortest AST. Mauchly's test did not show a violation of sphericity distribution of AST values (W=0.35, p=0.13). A one-way repeated measures ANOVA found a significant main effect of layout on AST (F(2,10)=94.1, $p < 0.05$). A post-hoc test with Bonferroni correction showed that the three layouts were significantly different from each other ($p < 0.05$ in all cases).

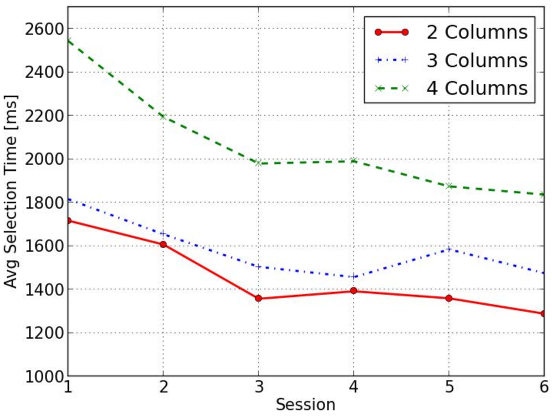


Figure 6. Grand mean for Average selection time (AST).

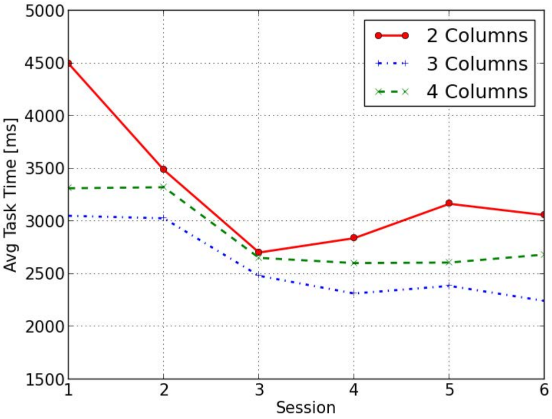


Figure 7. Grand mean for Average task time (ATT).

Regarding the Average task time, the grand mean for the six participants and the three layouts is presented in Figure 7. Interestingly, the 2C layout had the longest ATT, while the 3C layout had the shortest ATT. Mauchly's test showed a slightly violation of sphericity against layout (W=0.13, p=0.02), so we used the Greenhouse-Geisser correction method. A one-way repeated measures ANOVA with Greenhouse-Geisser correction (ϵ=0.54) found a significant main effect of layout on ATT (F(1.08, 5.4)=12.9, $p < 0.05$). A post-hoc test with Bonferroni correction showed that the 3C layout had a significant lower ATT than the other two layouts (2C and 4C), with $p < 0.05$ in both cases. Nonetheless, there was not significant difference between 2C and 4C (p = 0.15).

Discussion

As can be observed in Figure 5, the time needed to switch pages was about 3.5-4 seconds in the 2C and 3C layouts, raising up to 5 seconds in the 4C layout. Because the paging time was computed from the last selection within every visited page, this difference was expected since the participants tend to scan the context one last time before switching pages. Because in the 4C layout there were more columns to scan, paging in the four column layout was slower. APT was almost equal for 2C and 3C in the last two sessions. This might indicate that, in the 2C and 3C layouts, participants learned to scan a single column while relying on their peripheral vision for the adjacent columns, which could not be done in the 4C layout.

Table 1. Subjective impressions of the participants regarding speed and comfort

	2 columns	3 columns	4 columns
Perceived as faster	3	3	0
More comfortable	4	2	0

When we consider only the time employed to make a single selection with Context Switching, the AST for 2C was significantly faster than 4C, and a bit faster than 3C, as can be seen in Figure 6. Because the average distance between contexts increases with the number of columns, this result was expected, since longer saccades were needed to switch contexts in the 3C and 4C contexts. This result is also consistent with the participants interviews. As can be seen in Table 1, the 2C and 3C layouts received the same evaluation regarding perceived speed. Only one user perceived the four column layout as the second faster. The two columns layout was perceived as the simplest to use by all participants.

Comparing the time needed to make a single selection with the time to execute the meta-keys, it can be seen that selections were faster. This results is not surprising since meta-keys involve two saccades, while selecting with Context Switching requires a single one. Nonetheless, in applications like eye typing, the activation of meta-keys would less frequent than letter selection, for example, whenever the user needs to change the layout from characters to numeric or symbols and vice-versa.

The ATT metric reflects the overall performance of participants, considering both selection and paging. Figure 7 shows that the 3C layout had a significant shorter ATT than the other two layouts. Although the 2C layout had the shortest time to select one item (AST), a considerable number of paging operations were needed to browse all pages. On the other hand, the 4C layout had the smallest number of paging operations, however, the time to select one item was the longest among the three layouts. The 3C layout had a better balance between the number of selectable items on the screen and the number of paging needed to browse the collection entirely. This may indicate that balancing this two factors yields the better performance in tasks similar to the one tested in this experiment. One participant with previous experience in eye tracking said that with 3 columns it was easier to use peripheral vision to quickly explore a context and also was more comfortable to use than with the other two methods.

Subjective evaluation of meta-keys and Context Switching

Participants were asked about how easy it was to make selections using the interface and to execute the meta-keys. In a Likert scale from 1 (very hard) to 5 (very easy), the average response for selection was 4.7, i.e., the participants found it very easy to make selections using Context Switching. None of the participants complained about context resizing as disorienting, though in the four columns layout, there were some delayed responses due to implementation issues.

Activation of meta-keys in the vertical direction received a 2.8 of a maximum of 5 to scroll up, and 3.0 to scroll down, so that most participants found reasonable or good to use. In the horizontal direction, meta-keys made to the right side received a score of 2.4, while meta-keys made to the left were rated as 2.6.

The better scores for meta-selections in the vertical direction may just reflect that participants were able to learn them better, since activation of meta-keys in the vertical direction were required more often. Because the bridge between the markers and the contexts was relative small (1.5 cm), some users had to look out of the screen to make the selection, due to gaze tracking errors. We believe that by making the bridge a little wider, its activation should be facilitated since it will be more robust to gaze tracking errors.

Overall, the experimental results, as well as the interviews, indicate that meta-keys have the potential to extend the usability of gaze-based interaction. Because of the use of two gestures, meta-keys are less prone to accidental activation than single gaze gestures techniques [11]. At the same time, they cause less eye fatigue than more complex gestures, like in the EyeWrite [19] and Eye-S [14] interfaces. Similar to Context Switching paradigm [13], the use of a bridge between the interface and the markers reduces the effect of eye tracking noise and overall improve the robustness of the meta-keys detection. Further studies with more participants and a longer distance between the contexts and the markers should help to improve the use of meta-keys. Although implemented in a Context Switching application, the use of meta-keys can complement other interaction techniques, such as dwell time.

CONCLUSION

This paper introduced the concept of meta-keys with the objective to extend the functionality of gaze-based interaction. Meta-keys complement the interaction by executing less frequent commands, like changing the layout of the interface or browsing several views of the system. We defined meta-keys as two-step gaze gestures executed from within the interface to markers located around it. The space separating the interface from the markers acts as a bridge. To activate the meta-keys the user must cross the bridge two times: first from the interface to the marker, and then from the marker back to the interface. This reduces the number of accidental activations, avoiding the Midas touch problem and reducing the effects of the eye tracking noise. Extending the previous work of Tula et al. [16], we reported the results of a study where participants had to use meta-keys in a task of navigation and selection using three layouts with different number of keys. Results showed that participants learned the meta-keys easily and were able to use them to complete the tasks successfully. We found that to improve the overall performance and user experience, meta-keys should be less frequent than regular selection. Nonetheless, in some cases complex layouts can slow down the interaction, even if it implies a smaller number of meta-keys activation. Results showed that meta-keys are a promising tool for extending the functionality of gaze-based interfaces.

ACKNOWLEDGMENTS

This work has been supported by grants 2010/14075-4, 2012/04426-0, and 2011/00267-1, São Paulo Research Foundation (FAPESP).

REFERENCES

1. Drewes, H., and Schmidt, A. Interacting with the computer using gaze gestures. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, INTERACT'07, Springer-Verlag (Berlin, Heidelberg, 2007), 475–488.
2. Hansen, D. W., Skovsgaard, H. H. T., Hansen, J. P., and Møllenbach, E. Noise tolerant selection by gaze-controlled pan and zoom in 3d. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, ETRA '08, ACM (New York, NY, USA, 2008), 205–212.
3. Hansen, J. P., Tørning, K., Johansen, A. S., Itoh, K., and Aoki, H. Gaze typing compared with input by head and hand. In *Proceedings of the 2004 symposium on Eye Tracking Research & Applications*, ETRA '04, ACM (New York, NY, USA, 2004), 131–138.
4. Heikkilä, H., and Räihä, K.-J. Simple gaze gestures and the closure of the eyes as an interaction technique. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, ACM (New York, NY, USA, 2012), 147–154.
5. Huckauf, A., and Urbina, M. H. Gazing with peyes: towards a universal input for various applications. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, ETRA '08, ACM (New York, NY, USA, 2008), 51–54.
6. Isokoski, P. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, ETRA '00, ACM (New York, NY, USA, 2000), 15–21.
7. Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, ACM (New York, NY, USA, 1990), 11–18.
8. Lankford, C. Effective eye-gaze input into windows. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, ETRA '00, ACM (New York, NY, USA, 2000), 23–27.
9. Majaranta, P., Ahola, U.-K., and Špakov, O. Fast gaze typing with an adjustable dwell time. In *Proceedings of*

the 27th international conference on Human factors in computing systems, CHI '09, ACM (New York, NY, USA, 2009), 357–360.

10. Majaranta, P., and Räihä, K.-J. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, ETRA '02, ACM (New York, NY, USA, 2002), 15–22.
11. Møllenbach, E., Hansen, J. P., and Lillholm, M. Eye Movements in Gaze Interaction. *Journal of Eye Movement Research 6* (2013), 1–15.
12. Morimoto, C., Koons, D., Amit, A., Flickner, M., and Zhai, S. Keeping an eye for hci. In *Computer Graphics and Image Processing, 1999. Proceedings. XII Brazilian Symposium on* (1999), 171–176.
13. Morimoto, C. H., and Amir, A. Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, ACM (New York, NY, USA, 2010), 271–274.
14. Porta, M., and Turina, M. Eye-s: a full-screen input modality for pure eye-based communication. In *Proceedings of the 2008 symposium on Eye Tracking Research & Applications*, ETRA '08, ACM (New York, NY, USA, 2008), 27–34.
15. Räihä, K.-J., and Ovaska, S. An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 3001–3010.
16. Tula, A. D., de Campos, F. M. S., and Morimoto, C. H. Dynamic context switching for gaze based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, ACM (New York, NY, USA, 2012), 353–356.
17. Urbina, M. H., and Huckauf, A. Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, ACM (New York, NY, USA, 2010), 315–322.
18. Ward, D. J., and MacKay, D. J. C. Fast hands-free writing by gaze direction. *CoRR cs.HC/0204030* (2002).
19. Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 symposium on Eye Tracking Research & Applications*, ETRA '08, ACM (New York, NY, USA, 2008), 11–18.