Author(s):	Heikkilä, Henna
Title:	Gesturing with Gaze
Main work:	Proceedings of COGAIN 2008
Editor(s):	Istance, Howell; Štěpánková, Olga; Bates, Richard
Year:	2008
Pages:	43-46
ISBN:	978-80-01-04151-2
Discipline:	Computer and information sciences
Item Type:	Article in Conference Proceeding
Language:	en
URN:	URN:NBN:fi:uta-201311061564
URL:	http://www.cogain.org/cogain2008/COGAIN2008-Proceedings.pdf

Gesturing with Gaze

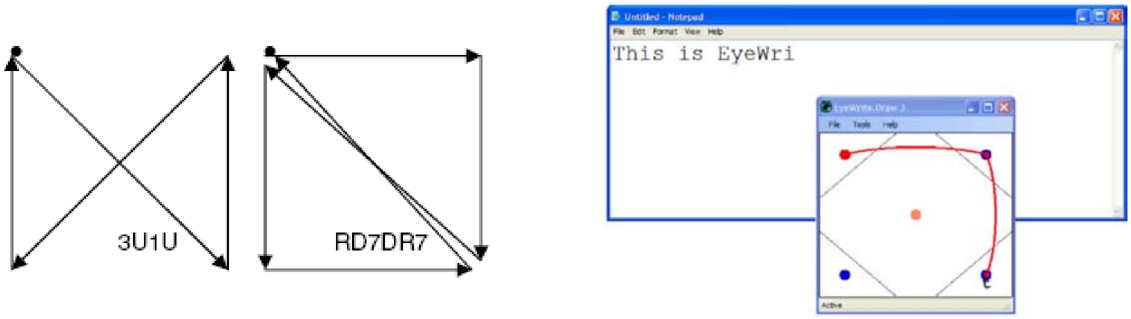
Henna Heikkilä
TAUCHI
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
henna.heikkila@cs.uta.fi

Keywords
Gaze-based interaction, drawing application, gaze gestures

Introduction

This paper describes the first experiment of a PhD thesis project studying the use of gaze and gaze gestures in drawing applications. Drawing programs is an ideal platform for gaze gestures, since they can save screen space when fewer buttons or menus are needed in the application. Gestures also provide an easy access to a selection of often-used tools, for example copy, paste or undo. In some cases, gestures can even be performed faster than a selection from a menu. Some gaze-controlled drawing applications have already been implemented. However, none of these uses gaze gestures. EaglePaint (Gips and Olivieri, 1996) is one of the first drawing applications for gaze. In EaglePaint, a randomly colored line appears wherever the user looks. Hornof et al. (2004) have designed a drawing application called EyeDraw. In EyeDraw, the user draws pictures by using lines, squares, circles, and some predetermined shapes.

Gestures are mostly used in mouse and pen-based interfaces, but recently gaze researchers have become interested in them as well. Drewes and Schmidt (2007) and Istance et al. (2008) used gaze gestures to control an application. Drewes and Schmidt designed a set of gaze gestures that were based on eight directions (cardinal and half-cardinal points). See two example gestures in Figure 1. They implemented the gestures in a media player and used the gestures to play, pause and stop a track, to change a track or a media channel and to adjust the volume. They noted that the participants were able to do the asked gaze gestures and they used the whole screen to perform the gestures, which was reported to be easier than to perform them in small scale. Istance et al. added gaze-based interaction to Second Life and used gaze gestures to switch between different types of gaze and mouse controls.



This is EyeWri

Figure 1. Example gestures Drewes and Schmidt's (2007) study (on left) and a screen caption from Wobbrock et al.'s (2008) EyeWrite (on right).

In addition to controlling an application, gaze gestures have also been used for writing with gaze. Wobbrock et al. (2008) implemented an application called EyeWrite (see Figure 1) that is based on their EdgeWrite method (for pen-based interaction). The shapes of the gestures in EyeWrite resemble roman letters so that they would be easier to learn and to remember. The gaze gestures are issued on a special writing window that shows guides to help the gesture drawing process. In their longitudinal study, Wobbrock et al. found that though the participants were slower when writing with gaze gestures than with dwell-based on-screen keyboard, they felt that the gaze gestures were an easier and faster way to write and caused less fatigue than the on-screen keyboard.

Another gaze writing style that resembles gaze gestures should be mentioned here. Isokoski (2000) designed a gaze writing prototype that is based on off-screen targets and a text input method called Minimal Device Independent Text Input Method (MDITIM). In MDITIM, the user writes using four principal cardinal directions (North, East, South, and West) and, in this prototype, these directions are marked on the monitor. The user writes by looking at the off-screen targets in a certain order. For example, when writing the letter y, the user looks at the off-screen targets in following order: South, East, South, and West. The letters create gesture-like gaze paths that the user can memorize.

Before designing gestures for a drawing application in detail, it is useful to know the usability of gestures in general. Usability here refers to users' ability to produce gestures and the system's ability to recognize them. As a first step, we carried out a controlled experiment.

Experiment

The goal of the experiment is to find out what kind of gaze gestures are easy and natural for the user. In this experiment, different shapes (such as lines, triangles, rectangles and circles) were presented to the participant and they tried to imitate them with their eyes.

Setup

For the test, 16 participants, 7 males and 9 females, were recruited. Their ages varied from 19 to 33 years. Four of the participants wore eyeglasses. Tobii 1750 eye tracker (with screen resolution of 1280x1024 pixels) and ClearView 2.6.3 were used during this experiment to track the participant’s eyes and to record the gaze data. The calibration was done with 16 points. With calibration, a test took about 30 minutes.

During the test, participants were asked to accomplish a set of tasks as fast and accurately as they could. Three of the tasks were done on an empty drawing area and another three tasks were done on a drawing area that had a model image in it. The tasks were:

- Draw with your gaze the letter L.
- Draw with your gaze a triangle.
- Draw with your gaze a line that goes from left to right and return to the starting point.
- Draw the form of the green rectangle (by looking at each of its corners one by one) with your gaze.
- Draw the form of the green circle (by following the circumference of the circle) with your gaze.
- Look once at each end of the green line.

All tasks were done with both small (250x250 pixels) and large (1180x920 pixels) drawing area. All tasks were repeated five times. In total, each participant did 60 tasks. The task order was balanced between participants.

Preliminary results

The analysis of the data is still ongoing and only preliminary results and some notes from the experiment can be presented at this point. The aim is to finish the analysis by September 2008.

In most cases, the shapes drawn during the tests resemble the asked shapes, at least when a human examines them. However, it is much more difficult for an application to recognise the shapes when the letter L resembles more the letter V. Participants commented that it was difficult to draw the shapes on almost blank surface. Especially, when the starting and the ending point of the shape were the same (e.g. triangle), it was difficult to remember where the drawing of that shape started, since there were no cues on a blank surface. Some participants used the borders of the drawing area as a help when drawing triangles and horizontal lines. This strongly suggests that the user needs some guides when doing gaze gestures. EyeWrite already uses guides to help the user to hit the target areas of the drawing window (Wobbrock et al., 2008). Drewes and Schmidt (2007) also offered helping lines in their test application, but the participants did not use them. However, the participants did use the whole display area for their gaze gestures and probably used the sides and corners of the application window or the display as a help.

After the test, most participants reported that the circle was the most difficult shape to draw. They said they had to concentrate more when trying to follow the curving form of the circle (this means more fixations on the gaze path, see Figure 2). When the task completion times are compared, the task concerning the circle shape done on the large drawing area stands out since it took about one and a half times as much time to complete the task than the second slowest task (rectangle shape done on the large drawing area). However, when comparing circle and rectangle shape tasks done on small drawing area, the task completion times are about the same in both tasks. These results are similar to the results from Tchalenko’s (2001) studies. He discovered that curves are very difficult to draw because the user has to concentrate to achieve a better control of his eyes.

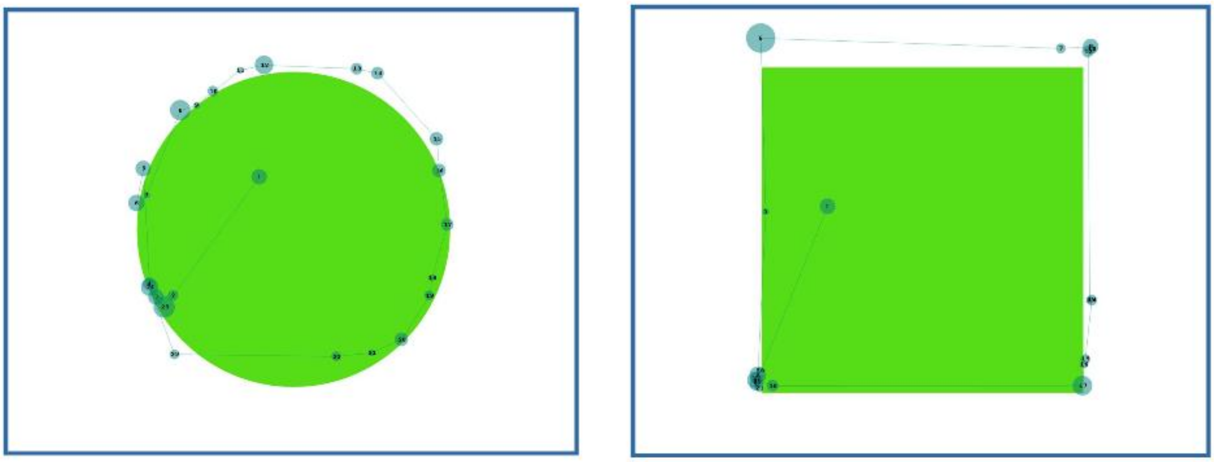


Figure 2. One participant’s gaze path, when trying to follow the circumference of the circle (on left) and when trying to hit the corners of the square. Both show the calibration inaccuracy problem.

Next steps

The analysis of the gathered data continues. First, the accuracy between the gaze path and the model will be calculated (for tasks done with a model). Though the calibration was done with 16 points and its result was excellent, the drawn gaze paths have some systematic inaccuracy, especially on the upper part of the display (see Figure 2). This shows how vulnerable gaze pointing is to calibration errors. Next, the gesture duration and the distance that the gaze travels when performing a gesture will be taken under examination (for tasks done on an empty drawing area). This step relates to a challenge that the gaze gestures face: how to determine when the gesture starts and when it ends? Afterwards, an algorithm that recognises the gaze gestures will be created.

Conclusion

The experiment described aims to shed light on which kind of gestures would be easy to do with gaze and which kind of algorithm is needed to recognise the performed gaze gestures. Results will help when designing a set of gaze gestures that are suitable for a drawing application and for its tools and functions. When the set of gaze gestures is ready, they will be tested with users and compared against other available methods.

References

Drewes, H. and Schmidt, A. (2007) Interacting with the Computer using Gaze Gestures. In *Proceedings of INTERACT 2007*. Lecture Notes in Computer Science 4663, Springer, pp. 475–488.

Gips, J. and Olivieri, P. (1996) EagleEyes: An Eye Control System for Persons with Disabilities. *The Eleventh International Conference on Technology and Persons with Disabilities.* Available in http://www.cs.bc.edu/~eagleeye/papers/paper1/paper1.html.

Hornof, A. J., Cavender, A., and Hoselton, R. (2004) EyeDraw: A system for drawing pictures with eye movements. *Proceedings of ASSETS 2004: The Sixth International ACM SIGACCESS Conference on Computers and Accessibility*. ACM Press, New York, NY, pp. 86–93.

Isokoski, P. (2000) Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (ETRA '00). ACM Press, New York, NY, pp. 15–21.

Istance, H., Bates, R., Hyrskykari, A., and Vickers, S. (2008) Snap clutch, a moded approach to solving the Midas touch problem. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (ETRA '08). ACM Press, New York, NY, pp. 221–228.

Tchalenko, J. (2001) Free-eye drawing. Point: Art and Design Research Journal, 11, pp. 36–41.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (ETRA '08). ACM Press, New York, pp. 11–18.