

A Gaze-Contingent, Acuity-Adjusted Mouse Cursor

Michael Dorr

University of Lübeck
Ratzeburger Allee 160,
D-23538 Lübeck,
Germany

dorr@inb.uni-luebeck.de

Christoph Rasche

Justus-Liebig-Universität
Giessen
Otto-Behaghel-Str. 10F,
D-35394 Giessen,
Germany

christoph.rasche@psychol.uni-
giessen.de

Erhardt Barth

University of Lübeck
Ratzeburger Allee 160,
D-23538 Lübeck,
Germany

barth@inb.uni-luebeck.de

Keywords

gaze-based input, gaze-contingent rendering, cortical magnification, human-computer interaction

Introduction

Gaze-based computer interfaces so far have often focused on using gaze as the sole input channel, for instance for eye-typing or for moving the mouse cursor. While such display control can be essential for disabled persons, it is generally rather awkward to use because gaze cannot be placed precisely on objects; furthermore, it cannot be used quickly enough for sequential input because of the *Midas touch* problem that both orienting and acting are transmitted through the same modality (Jacob, 1993). One solution to this problem can be not to rely on gaze exclusively, but to only use it as an assistive measure, for example to place the cursor at the gaze position only if mouse activity is registered (Zhai, Morimoto, & Ihde, 1999; Drewes & Schmidt, 2006). In the following, we present another such gaze-assistive approach; we address the problem that the position of a small mouse pointer is often forgotten during browsing, so that the pointer has to be relocated by a visual search before it can be used again. A common shortcut is that users wiggle the cursor briefly, thereby making it more conspicuous. To achieve a similar effect without the need of a manual intervention by the user, we render the mouse cursor in a gaze-contingent fashion and increase its size as a function of the distance from the centre of gaze. Because this size modification roughly follows the cortical magnification factor, the pointer has optimal visibility even in the user's visual periphery while keeping a normal-sized appearance when looked at directly. In some sense, this is the opposite of many previous gaze-contingent systems that made use of the reduced sensitivity of the visual periphery to leave out details, e.g. for compression or reduced rendering complexity (Perry & Geisler, 2002; Böhme, Dorr, Martinetz, & Barth, 2006).

Implementation

Many eye-tracking studies are tested on platform modifications not readily accessible to other users. We therefore decided to use open-source software – despite its much smaller share in the mass market – which lends itself naturally to niche and research domains because it can be flexibly modified and distributed to other researchers and users, and opted for the Linux operating system. We were then faced with the choice whether to write our own prototype software application, to adapt an existing application, or to make the gaze-contingent cursor be used system-wide. The first two options are appealing because they certainly require the least effort to implement; as a drawback, they are of fairly

little use outside the scope of this research project. Unfortunately, the last option would involve writing a complete proxy server for the X window system (Schneider, 1997), a somewhat daunting task. We therefore decided to settle on a middle ground and adapt the *Qt* cross-platform application and user interface framework (Trolltech, 2009). The advantage here is that with only minor modifications to a few well-encapsulated C++ classes in the *Qt* library, any application that is based on *Qt* (including the well-known KDE desktop system) can make use of the gaze-contingent cursor; by linking either against the modified or the original library, it is also possible to switch behaviour on a per-application level.

One of the central application of today’s desktop computers is the web browser. Unfortunately, both the KDE default browser *konqueror* and the *WebKit*-based browser that comes shipped as a demo with the *Qt* library implement their own cursor handling on top of *Qt*. Therefore, we modified the latter application as well.

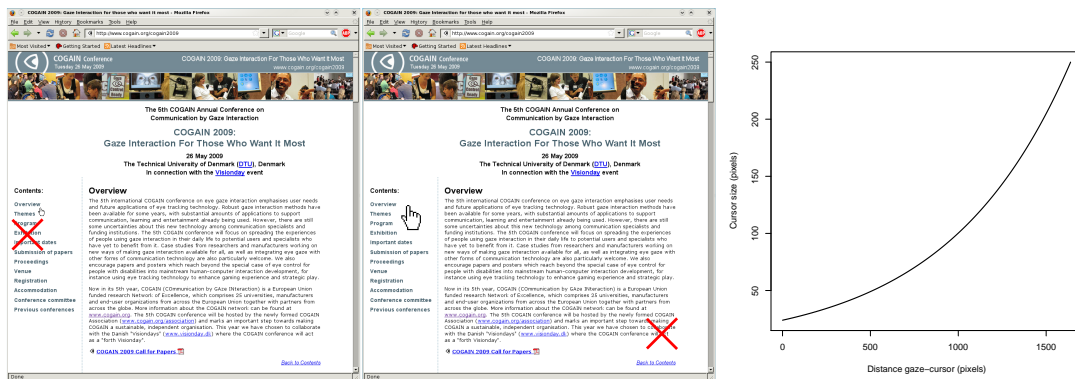


Figure 1. Schematic illustration of the gaze-contingent mouse cursor. Left: The subject is scanning the menu column to the left (gaze indicated by the red cross); the mouse cursor is small because it is still visible (para-)foveally. Middle: When the subject has read the text body and the distance gaze to cursor is large, the mouse cursor is enlarged to ensure visibility even in the visual periphery. Right: Cursor size as a function of distance between gaze and cursor.

Experimental validation

Four subjects, department members and undergraduate students, participated in a small usability study; all were naive regarding its purpose. The goal was to engage the subject in a task that required frequent scrolling within the windows and switching between windows in order to exploit the use of the gaze-contingent cursor as much as possible, given the limited duration of a typical evaluation (about 10 minutes). Subjects were asked to perform some typical web browsing activities such as looking up flight information, comparing prices for consumer electronics, etc. The web browser was placed on the right half of the screen and results of the tasks had to be entered into an editor window that covered the left half of the screen. During the experiment, eye movements were registered using an SMI iView X RED remote eye tracker running at 50 Hz. The size of the mouse cursor varied from 24 ($size_{min}$) to 250 ($size_{max}$) pixels according to the formula

$$size_{cursor} = size_{min} \cdot \left(\frac{size_{max}}{size_{min}} \right)^{d/d_{max}},$$

with d the current distance between gaze and cursor and d_{max} the maximum possible distance on the screen, i.e. the length of the screen diagonal (at a resolution of 1280 by 1024 pixels).

At the end of the test, subjects were asked to evaluate the use of the gaze-contingent cursor. They were asked to rate the gaze-contingent cursor on a scale from -5 (highly annoying) over 0 (neutral) to +5 (highly useful).

Results

None of the subjects rated the gaze-contingent cursor as annoying. In fact, one of the subjects only became aware of the nature of the cursor when she read the instructions to evaluate the system after the experiment session (this subject therefore did not assign a numerical rating). However, this may be explained by the fact that the tasks placed a high cognitive load on this subject, who had only very recently acquired the German language (all the web sites to be visited were in German). The remaining three subjects rated the cursor positively (+1, +1, and +4, respectively); two of them commented they “did not have to look for the cursor anymore”. Nevertheless, two subjects also reported that the cursor sometimes seemed to move by itself (presumably when jitter either in the eye or the gaze measurement induced a slight “pumping” effect, i.e. size variations), indicating the need for a more elaborate gaze data filtering.

Conclusion

We have presented a gaze-contingent mouse cursor that changes its size as a function of visual eccentricity to remain visible to the user even in the visual periphery. A small usability study showed that users were not distracted by the variable size, but rather rated the gaze-contingent cursor to be a positive addition to the user interface. A more objective performance evaluation will have to be carried out in the future to test whether this subjective preference also corresponds to a higher input efficiency. The presented idea could also be extended to editor cursors.

In order to implement the gaze-contingent mouse cursor, we have modified the publicly available *Qt* library; this opens up the possibility to use our cursor from a vast variety of different applications without the need to recompile them. The source code of our modifications is available upon request.

Acknowledgements

Our research has received funding from the European Commission within the project GazeCom (contract no. IST-C-033816) of the 6th Framework Programme. All views expressed herein are those of the authors alone; the European Community is not liable for any use made of the information. We acknowledge Henry Schütze’s assistance with programming and data collection.

References

- Böhme, M., Dorr, M., Martinetz, T., & Barth, E. (2006). Gaze-contingent temporal filtering of video. In *Proceedings of Eye Tracking Research & Applications (ETRA)* (pp. 109–115).
- Drewes, H., & Schmidt, A. (2006). Mauszeigerpositionierung mit dem Auge. In *Proceedings of Mensch & Computer 2006* (p. 185-94).
- Jacob, R. J. K. (1993). Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in Human-Computer Interaction*, 4, 151-80.
- Perry, J. S., & Geisler, W. S. (2002). Gaze-contingent real-time simulation of arbitrary visual fields. In B. E. Rogowitz & T. N. Pappas (Eds.), *Human Vision and Electronic Imaging: Proceedings of SPIE, San Jose, CA* (Vol. 4662, p. 57-69).
- Schneider, J. (1997). *X11-big-cursor minihowto*. "<http://i11www.ira.uka.de/~schneid/X11-big-cursor/master/>".
- Trolltech. (2009). "<http://www.qtsoftware.com>".
- Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* (p. 246-53).