S. Diefenbach, N. Henze & M. Pielot (Hrsg.): Mensch und Computer 2015 Tagungsband, Stuttgart: Oldenbourg Wissenschaftsverlag, 2015, S. 267-270.

Assisting Mouse Pointer Recovery in Multi-Display Environments

Florian Fortmann[1], Dennis Nowak[2], Kristian Bruns[2], Mark Milster[2], Susanne Boll[2]

OFFIS – Institute for Information Technology[1]
University of Oldenburg[2]

Abstract

Recovering the mouse pointer in a multi-display environment after it got lost, e.g., because the user shifted his or her visual attention to another task, can be very annoying and frustrating. The reason is that the user has to oversee a large display space requiring sequential sampling actions. These actions consume psycho-physiological effort for moving the eyes and head around, and increase the time needed to recover the mouse pointer. In this paper, we present an assistant system to support users to recover their mouse pointers in multi-display environments. A preliminary user study showed no significant improvement of mouse pointer recovery time, but usability and satisfaction were assessed positive. This work was carried out as part of a course on human-machine interaction at the University of Oldenburg.

1 Introduction

From time to time, when users are working with a mouse device, they fall prey to the "lost mouse pointer phenomenon". This phenomenon means that users sometimes simply do not know where their mouse pointers are on their displays, e.g., after they shifted their visual attention to other objects or tasks. Recovering a mouse pointer after it got lost in a multi-display environment, as it can be found, e.g., in mission monitoring and control rooms, can be very annoying and frustrating. A major reason is that users working in multi-display environments have to oversee large display spaces in the periphery of their visual field. This requires increased efforts for moving the eyes and head around for mouse pointer recovery, and increases the time needed to recover the mouse pointer. In recent years, using eye tracking systems to improve human-machine interaction has become a popular field of research. The reason can be seen in the progress that has been made in eye tracking technology in the last decade, e.g., with regard to accuracy of measures and comfort (Johansen et al. 2011). A major field of research focuses on using eye movements directly as interaction techniques – so called gaze-based interaction. For example, the benefits of using an eye tracking system for gaze-based target selection in a computer game were investigated in (Leyba & Malcolm 2004). In (Vickers et al. 2008) the tool SnapClutch was developed to overcome problems associated with the Midas Touch problem in context of multi-player games. Other studies have shown that a combination of mouse, keyboard and gaze-based interaction can improve human performance. For example, in (Zhai et al. 1999) the Manual And Gaze Input Cascaded (MAGIC) pointing technique was presented. It was shown that MAGIC reduced physical effort and fatigue as compared to traditional pointing with a mouse device.

In this paper, we present the mouse pointer recovery assistant system WIMM, which stands for “Where Is My Mousepointer?”. WIMM continuously analyses a user’s point of regard to provide optimal mouse pointer recovery assistance. Two assistance strategies were designed and evaluated in a preliminary test setting. The first strategy allows a user to call the mouse pointer to the position of his or her current point of regard. The second strategy displays an arrow in the user’s current visual focus pointing to the current location of the mouse pointer. Although our results did not indicate a significant improvement of mouse pointer recovery time by using WIMM, the results indicated that both assistance strategies were perceived usable and satisfactory. This work was carried out as part of a course on human-machine interaction at the University of Oldenburg.

2 Design and Evaluation

After we familiarized ourselves with multi-display environments, we performed a brainstorming with five participants to identify user requirements, e.g., reduction of mouse pointer recovery time and effort, and to envision possible design solutions (DS). In general, the participants agreed on an assistant system-based approach and favored the following five design solutions: (DS1) the mouse pointer continuously follows the visual focus of the user; (DS2) the mouse pointer moves to the user’s point of regard if requested; (DS3) tactile mouse feedback indicates the current location of the mouse pointer; (DS4) an arrow is displayed in the user’s point of regard to indicate the current location of the mouse pointer; (DS5) an ambient light display indicates the current location of the mouse pointer. Then, we performed an online questionnaire, where we explained DS1-DS5 and asked designated users to vote for their preferred design solutions. The questionnaire was completed by 38 participants aged between 18 and 45 years. Multiple selections were possible. The results indicated that DS2 (15 votes) and DS4 (14 votes) were preferred over DS1 (9 votes), DS5 (6 votes), and DS3 (2 votes). Based on this feedback we decided to further investigate the design solutions DS2 and DS4. For both design solutions, the assistant system needs information about the current position of the mouse pointer within the multi-display environment, and the user’s current point of regard, which can be determined by modern eye tracking systems. The foundation of our prototype was an Ergoneers Dikablis eye tracking system. This eye tracking system uses a head unit to determine the point of regard of a user within pre-defined areas of interest in real-time. In contrast to most remote eye tracking systems, an exact mapping of the point of regard to display coordinates is not supported by the used eye tracking system. For the prototype development we had to take this technical constraint into account. Therefore, each display of the multi-display environment was split into nine areas of interest (AOI). DS2 was implemented as follows: pressing the keys "Win+A" of the keyboard lets the mouse pointer move to the central area of the user's current display of regard. DS4 was implemented as follows: after the display containing the mouse pointer is neglected for more than 10 seconds, an arrow was automatically displayed in the central area of the user's current display of regard. The arrow points to the area containing the mouse pointer. The arrow was colored in green, yellow, or red to indicate the Euclidean distance between the AOI of regard and the AOI containing the mouse pointer. Our final design solutions are shown in Figure 1.

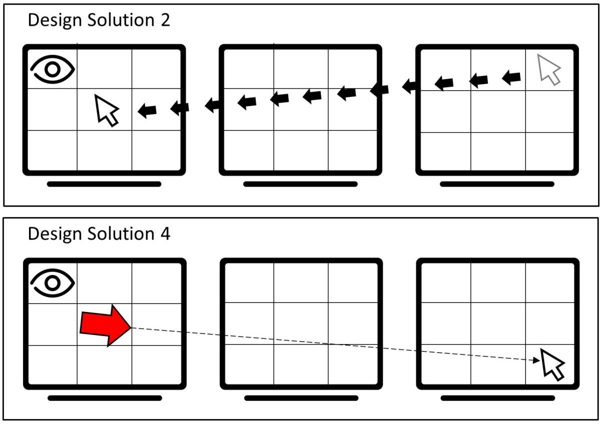

Figure 1: Implemented design solutions.

An experiment was conducted to analyze whether WIMM reduces mouse pointer recovery time. We invited 10 participants aged between 18 and 39 years. No participant had previous experience with eye tracking systems or assistant systems for mouse pointer recovery. Our multi-display environment consisted of 3 24-inch displays (left, center, and right). Each experiment session lasted for approx. 60 minutes. Each participant performed 30 repetitions of two scenarios. The first scenario was dedicated to DS2, and the second scenario was dedicated to DS4. Each scenario was performed without WIMM (control condition) and with WIMM (experiment condition). In each scenario, we counterbalanced the order of conditions to avoid sequence effects. In the DS2 scenario, the participants had to recover the mouse pointer after it was beamed to a random position on one of the three displays (start of measure) and then press a button close to the user's initial point of regard (end of measure). In the DS4 scenario, the participants had to recover the mouse pointer after it was beamed to a random position on one of the three displays (start of measure) and then press a button, which was located close to the position where the mouse pointer was beamed (end of measure). For each scenario we measured the time needed by the participants to finish the tasks. At the end of each session, we asked for qualitative feedback.

In the DS2 scenario, the participants needed in average less time (M = 2213 ms, SD = 793.86 ms) to complete the task without WIMM than with WIMM (M = 2529 ms, SD = 1163.58

ms). A Student's t-test showed no significant difference between these two conditions ($p = 0.08$). In the DS 4 scenario, the participants needed in average less time ($M = 2494$ ms, $SD = 874.78$ ms) to complete the task without WIMM than with WIMM ($M = 2764$ ms, $SD = 1384.57$ ms). A Students's t-test showed no significant differences between these two conditions ($p = 0.24$). Although the analysis of mouse pointer recovery time was somewhat disappointing for us, we received very positive qualitative feedback. All participants reported that WIMM was perceived usable and satisfactory.

3 Conclusions

We assume that there are different explanations for our results. First, we performed the experiment in an environment with three displays. This is possibly a too small display space to get the results we expected. We know that there are professional work environments, where humans have to supervise complex processes on larger display spaces, e.g., in nuclear power plants. Second, more training is possibly needed. One can also ask whether the reduction of mouse pointer recovery time is the ultimate measure to be taken into account. We mentioned that we also expected a decrease of effort by using WIMM. In our study, we did not measure effort. Further, it may also be a benefit of WIMM, if the act of recovering the mouse pointer becomes more acceptable – or even turns it into a playful and joyful activity. As reported by users, using WIMM was perceived usable and satisfactory. Overall, we believe that our results warrant further research on the WIMM prototype. For example, it would be very interesting to test WIMM in real working environments with larger display spaces, or with a remote eye tracking system.

Literature

Johansen, S. A., San Agustin, J., Skovsgaard, H. & Hansen, J. P. & Tall, M. (2011). Low cost vs. high-end eye tracking for usability testing. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*. New York: ACM.

Leyba, J. & Malcolm, J. (2004). Eye tracking as an aiming device in a computer game. In *Course work (CPSC 412/612 Eye Tracking Methodology and Applications by A. Duchowski)*. Clemson: Clemson University.

Istance, H., Bates, R. & Hyrskykari, A. & 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)*. New York: ACM.

Zhai, S. & Morimoto, C. & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)*. New York: ACM.

Contact

Florian Fortmann (florian.fortmann@offis.de)
OFFIS – Institute for Information Technology
Escherweg 2, 26121 Oldenburg