The MAGIC Touch: Combining MAGIC-Pointing with a Touch-Sensitive Mouse

Heiko Drewes[1], Albrecht Schmidt[2]

[1]Media Informatics Group, LMU University of Munich,
Amalienstraße 17, 80333 München, Germany
heiko.drewes@ifi.lmu.de

[2]Pervasive Computing Group, University of Duisburg-Essen,
Schützenbahn 70, 45117 Essen, Germany
albrecht.schmidt@acm.org

Abstract. In this paper, we show how to use the combination of eye-gaze and a touch-sensitive mouse to ease pointing tasks in graphical user interfaces. A touch of the mouse positions the mouse pointer at the current gaze position of the user. Thus, the pointer is always at the position where the user expects it on the screen. This approach changes the user experience in tasks that include frequent switching between keyboard and mouse input (e.g. working with spreadsheets). In a user study, we compared the touch-sensitive mouse with a traditional mouse and observed speed improvements for pointing tasks on complex backgrounds. For pointing task on plain backgrounds, performances with both devices were similar, but users perceived the gaze-sensitive interaction of the touch-sensitive mouse as being faster and more convenient. Our results show that using a touch-sensitive mouse that positions the pointer on the user's gaze position reduces the need for mouse movements in pointing tasks enormously.

Keywords: Eye-tracking, eye-gaze pointing, touch-sensitive mouse, MAGIC pointing.

1 Introduction

Pointing and selecting are very important operations in graphical user interfaces (GUIs). Typical input devices for pointing are mouse devices, trackballs, touch pads, or touch screens. Studies on the performance of these input devices started 30 years ago [3]. Research on eye-tracking systems for controlling computers with eye-gaze started in the 1980s [2]. Using an eye-tracker as a pointing device is an obvious approach utilizing eye-gaze for computer input.

Eye-tracking technology exists now for many decades and eye-trackers matured over the last years [4]. Current video-based eye-tracker systems work unobtrusively and provide also head-tracking, so that the users can move their heads freely while working with the system. Since such eye-tracking systems consist mainly of video

cameras and software, eye-tracking technology will fall in price and may be included in future systems without significant additional hardware costs. The high costs for an eye-tracker in the past made such systems only affordable in the field of accessibility. Consequently, existing eye-tracker applications for computer input use eye-gaze as the only input modality. A typical example is eye-typing for quadriplegics. In contrast, we focus on the use of eye-gaze as an additional input modality for the masses. The main challenge is figuring out how to integrate eye-gaze as an additional modality and to find forms of interaction, which are appropriate and useful. Although Bolt [2] gave a vision of gaze-aware multimodal interfaces already in 1981, an interface working similar to human-human interaction using gaze and speech is still a topic of research [17] and far away from application in real life.

Our research starts with the simple idea to position the mouse pointer at the point where the user looks at when she or he touches the mouse. We enhance Zhai's MAGIC (Mouse And Gaze Input Cascaded) pointing [16] by combining it with a touch-sensitive mouse device, similar to the mouse presented by Hinckley [7].

1.1 Eye-Gaze Interfaces

Jacob [8] did the first systematical research on eye-tracking as a pointing device in GUIs. His research investigated basic gaze interaction techniques like selection, scrolling text, and choosing menu items from pop-up menus. He identified the problem of accidental invocation by inspection and called it the Midas-Touch problem. Eye-tracking interfaces typically use the dwell-time method to avoid the Midas Touch problem. In this method, the user has to look for a certain time at a particular position on the screen to activate a function. This approach has been proven to be very useful for disabled users [10]. However, normal users work more efficiently with standard interfaces, e.g. mouse and keyboard. The dwell time required to activate the function nullifies the speed benefits resulting from the proverbial fast eye movements. Typically, dwell time adds about 500 to 1000 milliseconds to the overall interaction time [11], and results in pointing tasks that take longer than with a traditional mouse. Shortening the dwell time is not a viable solution, since it results in an increase in accidental function invocations by just looking at the screen. Another possibility to solve the problem is the use of a gaze key. Pressing the gaze key triggers the action belonging to the GUI object the user is looking at. As shown in [15], the use of a gaze key also speeds up pointing tasks.

When using eye-tracking systems, pointing accuracy is a central issue. The eye-tracker system used in our experiments offers an accuracy of ±0.5° visual angle, which results in minimum target sizes of about 36 pixels on current desktop systems. In contrast, current window-based GUIs typically require a precision better than 16 pixels to select a menu item and an even higher precision to position a text cursor between two letters. The accuracy of the gaze position is not only a question of the eye-tracker quality, but also of the intrinsic nature of eye movement. The size of the fovea (the high resolution area of the retina) is about one degree and to see something clearly the eyes move so that the projection of the object of interest falls inside the fovea but not necessarily in the center of it. Hence, better hardware or algorithms cannot easily improve the accuracy of eye trackers. As a result, user interfaces

designed for gaze interaction have to accommodate a larger threshold of pointing inaccuracy. Recent efforts have tried to solve the accuracy problem by adding intelligence to gaze positioning [14]. An intelligent algorithm positions the mouse pointer within the radius of inaccuracy at the target the user intended to activate. Other approaches use expanding targets [12] or enlargement of the region for eye-gaze input [9].

A further challenge for eye-gaze interfaces is the provision of feedback. In general, it is difficult to work with a system that does not provide visual feedback. A gaze pointer, analogous to a mouse pointer, does not work well because even little inaccuracies (e.g. calibration shift, intrinsic inaccuracy) set the gaze pointer next to the position where the eye is looking. When the eye tries to look at the gaze pointer the pointer moves again and this ends up in chasing the pointer across the screen. Typically, eye-trackers do not report small shifts of the eye gaze position as a consequence of noise filtering. This avoids chasing the pointer but leads to a jumping pointer hopping across the targets.

1.2 Pointing Devices

Currently used input technologies, i.e. keyboard and mouse, require mechanical movement and hence cause problems for some users. One common problem is that people after prolonged usage can suffer from the carpal tunnel syndrome. With the current trend towards bigger displays, the mouse movement distances will also increase and consequently the physical load on the hand muscles.

Additionally, people can have problems finding the position of the mouse cursor on the display [1]. There are several solutions to this problem found in commercial GUIs. One solution is a highlighting of the mouse pointer; another solution is to position the mouse cursor at the position where the user expects it. For example, the common Windows GUI offers two options in the control panel: (1) Show location of pointer when I press the CTRL key, (2) Automatically move pointer to the default button in a dialog box, and (3) the API knows the DS_CENTERMOUSE parameter for creation of dialogs. These mechanisms are designed to ease the use of the mouse and show that there is a potential area for improvement.

One approach that inspired our work is Zhai's MAGIC (Mouse And Gaze Input Cascaded) pointing [16]. The basic idea of the MAGIC pointing is positioning the mouse cursor at the gaze position on the first mouse move after some time of inactivity. The concept of MAGIC pointing is to use the gaze position for coarse positioning and the mouse for fine positioning. One problem of MAGIC pointing is the effect of overshooting the target because the hand and mouse are already in motion at the moment of pointer positioning. Zhai et al. suggests compensating this problem by taking the distance vector to the target and the initial motion vector of the mouse move into consideration when calculating the pointer position. In contrast, our research started with adding a touch sensor to the mouse and using the mouse to detect touch instead of movement. This approach does not require any compensation techniques, as the mouse does not move at the moment of pointer positioning. It also has the advantage that the user can invoke the gaze positioning at any time instead of waiting for the mouse inactivity timeout to expire.

2 The Touch-Sensitive Mouse

The work of Hinckley [7], which uses a touch-sensitive mouse to hide and display menus, inspired the initial design of the touch mouse. Our first naïve idea was to position the mouse pointer at the moment the users touches the mouse. Consequently, in our first prototype the entire mouse was touch-sensitive. Observation during a pilot study showed that users tend to leave their hand on the mouse while looking for the next target to click. Therefore, we created a prototype where only the left mouse button is touch-sensitive (see Fig. 1). When the finger contacts the mouse key (rising edge of the touch signal), the system sets the mouse pointer at the gaze position reported from the eye tracker. This gives the user the possibility to check the pointer position before performing the click. As long as the finger stays on the touch sensor, the mouse behaves like an ordinary mouse.

Fig. 1. The pictures show the first and second prototype of the touch-sensitive mouse.

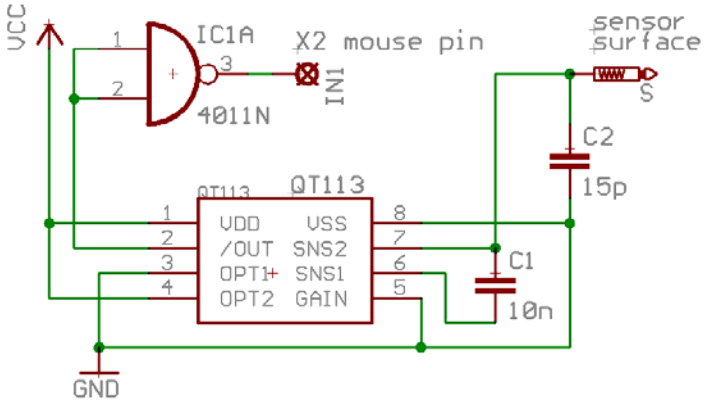

Fig. 2. Schematic of the circuit integrated into the mouse to detect touch.

The sensor for detecting touch uses capacitive sensing. The recognition time for the touch is critical, because the time interval between touching the sensor and

performing a click can be very short. The first sensor chip we used turned out to be to slow as it reported the touch signal after the mouse click. Therefore, the sensor chip used in the project is a capacitive sensor (QT 113 [13], see Fig. 2 for the schematic) with a very fast response time of 30 milliseconds. We connected the output of the touch sensor chip to the X2 button of the mouse. The X2 button has an assigned meaning in the Windows operating system and generates an event by the interrupt-driven mouse driver. This helps to speed up the positioning process, which is critical for the user experience. The overall reaction for touch positioning of the pointer is in the order of 60 ms (XP OS time slice 10 ms, touch detection 30 ms, displaying on TFT screen 20 ms), which appears to the user as an "immediate" reaction.

3 The User Study

In a user study, we compared the pointing performance for a traditional mouse and the touch-sensitive mouse with gaze positioning. We used the commercial eye tracker ERICA [5] and software developed for the study.

We recruited 10 participants aged 23 to 46 years old. All users were regular computer users and familiar with a mouse device.

We utilized a within-subject design with two independent variables: input modality (traditional mouse or gaze position using touch-sensitive mouse) and background image (plain background or complex background, see Fig. 3). The dependent variable was the duration of a pointing task. The condition of a complex background was introduced, because it is reasonably good simulation of a situation where the user cannot find the mouse pointer. We used a fixed target size of 100 pixels (3°) for the experiment. Pilot studies showed that Zhai's concept of raw and fine positioning also works with the touch-sensitive mouse, and we wanted to focus on the triggering by touch and not go into a discussion of Fitts' law for the eye. The distances between the starting position of the pointer and the target were chosen at random with an average distance of 400 pixels. In each condition, we asked the participants to do 50 pointing tasks. For each task, they had to move the pointer over the target and make a left click.

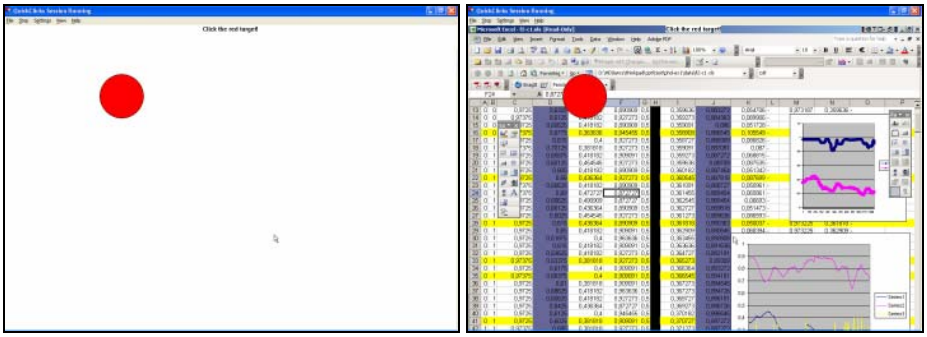

Fig. 3. The two background conditions. On the complex background, the mouse pointer is more difficult to find.

At the beginning of the experiment, users received a one-minute explanation of the basic function of the system and then had some time (~1 min) to tryout the system and familiarize themselves with its functions. Then the participants carried out the pointing tasks for all four conditions, with short breaks in between. The first task was on a white background using a traditional mouse. In the second task, the participants used the touch-sensitive mouse. Afterwards, we asked the users with which input method they performed faster and which one is more convenient. The next two runs were a repetition of the first two runs but on complex background. After the experiment, we asked the participants again, which method they perceived as being faster and more convenient. We did not randomize the order of the experiments because we wanted to have the answers for the blank background condition first (see 4.1). As shown in the discussion later, the picture for the interaction speed with the touch-sensitive mouse is quite clear. We do not expect to get further insights by a repetition of the user study with randomized task order.

4 Results and Discussion

We took the medians from the 50 click of a task to avoid problems with outliers. Table 1 summarizes the results. It is easy to see that there is no speed benefit for gaze positioning on blank background but for complex background.

Table 1. Medians for the total time in milliseconds for each of the participants on blank and complex background and the resulting arithmetic mean over the entire group.

	blank background		complex background	
Participant	mouse positioning	gaze positioning	mouse positioning	gaze positioning
P1	1097.0	751.0	1382.0	826.5
P2	971.5	1007.0	1066.0	806.0
P3	1121.0	716.0	1276.5	791.0
P4	1096.5	1066.5	1352.0	896.5
P5	932.0	871.0	1191.0	696.0
P6	926.5	1256.5	1066.0	1121.0
P7	1111.0	957.0	1217.0	891.0
P8	1211.5	1327.0	1412.0	1327.0
P9	1062.0	1482.0	1266.5	1277.0
P10	976.0	881.0	1101.0	656.0
Mean	**1051**	**1032**	**1233**	**929**
Std. Dev	94	253	128	234

The values in Table 2 are the results of paired Student's t-test for four different combinations and give the probabilities that the compared data sets are from a

distribution with the same mean value. There is strong significance that gaze positioning is faster when using a background, i.e. the user is not aware of the mouse pointer position. The values also show that mouse positioning takes longer on a complex background. The effect of better performance for gaze positioning on a background compared to no background (929 v. 1032 ms) is most likely an effect of learning due to not randomizing task order.

Table 2. T-tests for four different task combinations.

gaze vs. mouse pointing		background vs. blank background	
blank background	with background	gaze positioning	mouse positioning
0.823686	**0.002036**	**0.020738**	**0.000014**

4.1 The Subjective Speed Benefit on Blank Background

In the post-study interview, all but two users felt they were faster with the touch-sensitive device on the blank background, and all users felt they were faster with the touch-sensitive device on the complex background. However, the data show that their perceptions are only valid for the complex background. For the blank background, their completion times were equal.

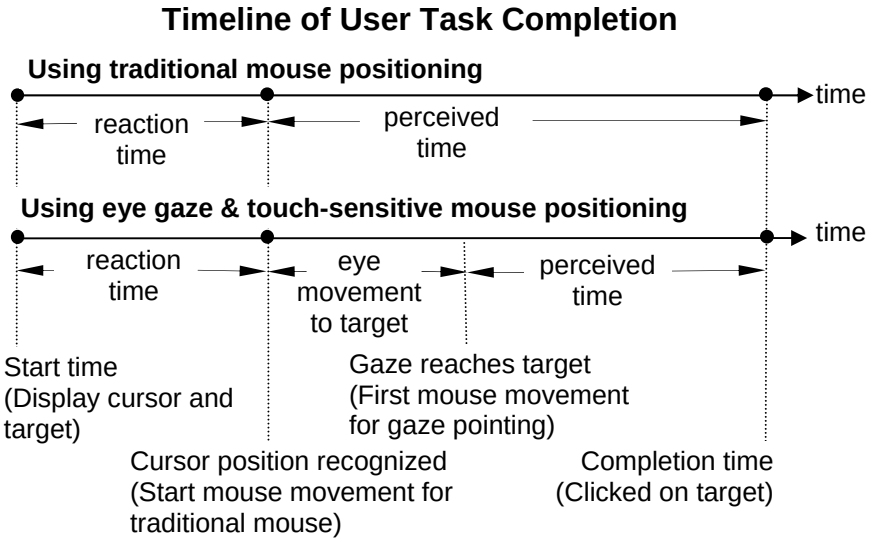

Fig. 4. Explanation of perceived speed-up with a touch-sensitive mouse and gaze positioning on a blank background.

When setting up the user study we experienced the phenomenon ourselves and for this reason, we put the question on subjective speed in the interview. Because we did

not want to bias the participants with the task on complex background, where the gaze positioning brings a real speed benefit, we decided to start with the task on blank background for all participants.

Fig. 4 illustrates a potential explanation for users' perception that the touch-sensitive mouse is faster. It shows the steps for task completion on a timeline for the condition with a blank background. Even though the overall time to complete the task is the same for classical pointing and gaze positioning, the time it takes users to move their hands is shorter when using gaze. This suggests that the users do not perceive the time required moving the eye as time needed to perform the task. Thus, users experience the use of eye-gaze as faster than a traditional mouse.

4.2 A Real Speed Benefit on Blank Background

Fig. 5 plots the time measured for both pointing task against the index of difficulty ($\log_2$(distance/target size)), as typically done for the evaluation of Fitts' law [6]. As expected, the plot for the classical mouse task shows a dependency of time from the distance between target and mouse pointer. This is in accordance to the experiments of Card [3]. In contrast, there is no dependency on the distance for the positioning by eye gaze. The explanation here is that the only task for the eye is to find the target and the only task for the hand is to click – the position of the mouse pointer does not matter for completing the task. The average distance of pointer and target of about 400 pixels (12°), given by the dimensions of the display used, was the reason that the completion time for the task with blank background was the same for classical mouse and touch mouse. This is the distance where both methods need the same time. It is the crossover-point of the constant function for the touch-sensitive device and the monotonic increasing function (Fitts' law) of the classical device. For shorter distances, the time to check whether the touch of the sensor positioned the mouse pointer correctly exceeds the time of a mouse move. For bigger distances, moving the mouse takes longer than the constant time of gaze positioning. As a consequence, we expect that there will be a speed benefit for the touch-gaze setup for longer distances, e.g. on very large screens or multi-screen setups.

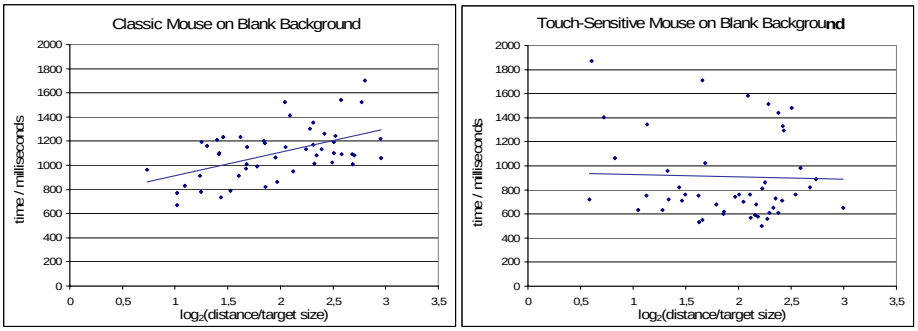

Fig. 5. Typical plots for the time to click the target against the index of difficulty for classical mouse and gaze positioning.

4.3 The Speed Benefit on Complex Background

Every click-task started with displaying target and pointer. On blank background, the participants spotted both objects with pre-attentive perception. The data show mouse pointer and gaze move independently towards the target where they meet. Fig. 6 depicts a mouse trail and a gaze path. The dotted gray lines connect points for the same time on mouse and gaze trail. It is easy to see that the gaze is already at the target when the mouse only just started moving towards the target.

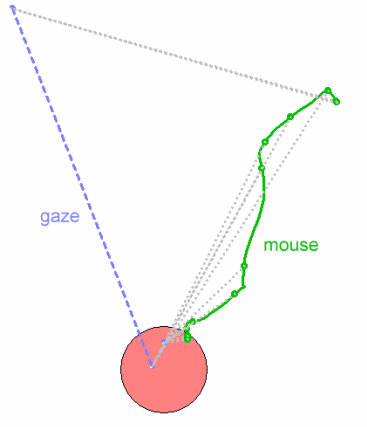

Fig. 6. Gaze (dashed) and mouse trail (solid) for the classical mouse task without background. The dotted grey lines connect points of same time.

Fig. 7 shows the corresponding plot for the distance of gaze and mouse pointer to the center of the target over time. The gray area visualizes the size of the target

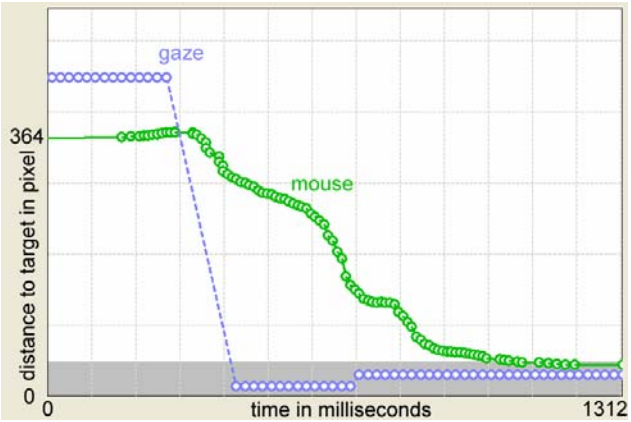

Fig. 7. Gaze (dashed) and mouse trail (solid) for the classical mouse task without background as a plot of distance to target over time.

On a complex background, it is possible to spot the big red target but not the mouse pointer. All participants started to stir the mouse to create a visual stimulus for finding the pointer. No participant moved the focus of the gaze onto the pointer; peripheral vision of movement is sufficient to locate and direct the mouse pointer. When using gaze pointing with the touch sensitive mouse the position of the mouse pointer has no relevance and consequently, it is not necessary to stir the mouse. The speed benefit for gaze positioning on complex background results from saving the time to stir the mouse.

Fig. 8 shows a typical example of pointing on a complex background and Fig. 9 gives the corresponding plot for the distance over time.

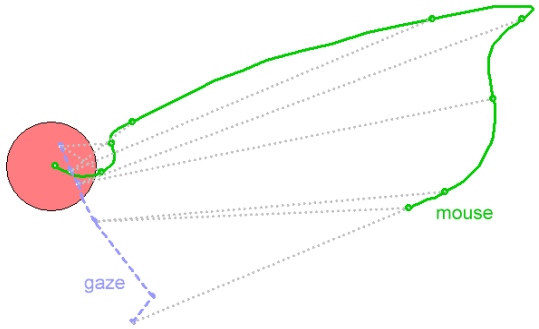

Fig. 8. Gaze (dashed) and mouse trail (solid) for the classical mouse task on complex background. The user stirs the mouse to detect its position by movement.

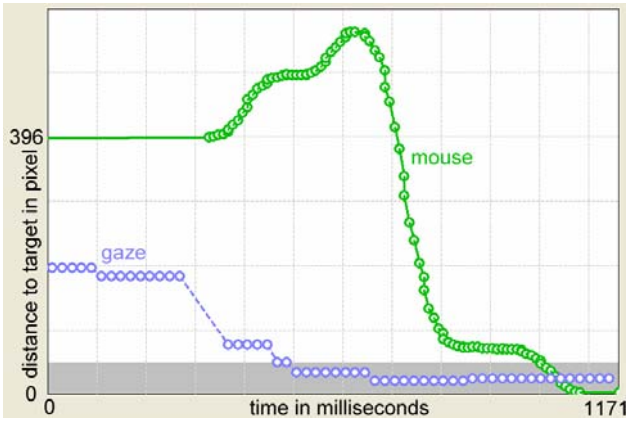

Fig. 9. Gaze (dashed) and mouse trail (solid) for the classical mouse task on complex background as a plot of distance to target over time.

It is worth to mention that for both background conditions the gaze arrives earlier at the target than the mouse. Nevertheless, with modern mouse accelerator techniques the mouse can be temporarily faster than the eye (see slopes in Fig. 9).

4.4 Reduction of Mouse Movements and Workload for the Eyes

It is obvious that gaze positioning saves an enormous amount of mouse moves. It was a surprising observation during the user study that the participants did not move the mouse at all. In case of missing the target, they did not move the mouse. Instead, they lifted the finger and repositioned the mouse pointer again by gaze.

It seems also obvious that using gaze positioning does not put extra workload to the eyes because the eyes move to the target anyway. It is hard to imagine that we can click a target without looking at it. However, a closer look to the data for the classical mouse tasks revealed that in approximately 20% of the recordings the eye gaze did not hit the target, but only moved close to it. In other studies, we observed that, when offering big targets, the eye gaze does not move to the center of the target, but already stops after slightly crossing the edge of the target. However, the end point of the gaze trail in Fig. 10 is definitely too far away from the target edge for an explanation with accuracy and calibration errors. Still in 6.7% of all classical mouse click tasks the eye gaze moved less than 90% of the distance from the initial gaze position to the target edge. This leads to the conclusion that it is possible to hit a target without looking at it.

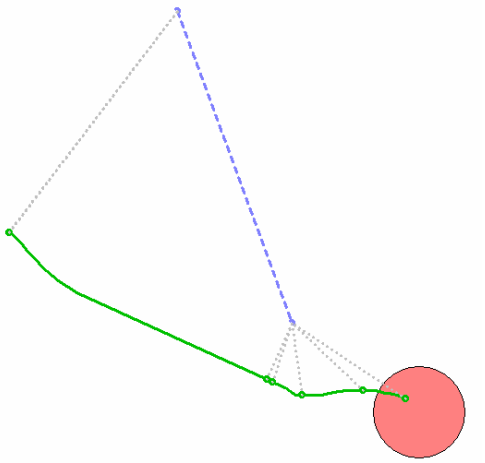

Fig. 10. Gaze (dashed) and mouse trail (solid) for the classical mouse task without background. Here the eye gaze does not hit the target.

Attempts to reproduce such data intentionally showed that this is indeed possible. The target used is big enough and its red color is striking, which allows easy detection by peripheral vision. The position of the standard mouse pointer is less obvious but can be easily located with peripheral vision because of its movement. As a consequence, it is possible to click on the target without directly looking at it. Nevertheless, such cases do not really contribute to the distance covered by the gaze and therefore it is correct to state that gaze positioning does not require extra gaze activity.

4.5 User Experience

After the two runs on blank background and after the two runs on complex background, we asked the participants to respond to the statement: “It is convenient not having to search for the mouse pointer.” The participants answered using a Likert scale from 1 (completely agree) to 5 (completely disagree). Fig. 11 presents the results. After the tasks on the complex background, agreement shifted towards “completely agree”. In the discussion after the experiments, the majority of the users mentioned that the appearance of the mouse pointer at the where you are looking seems very natural and intuitive.

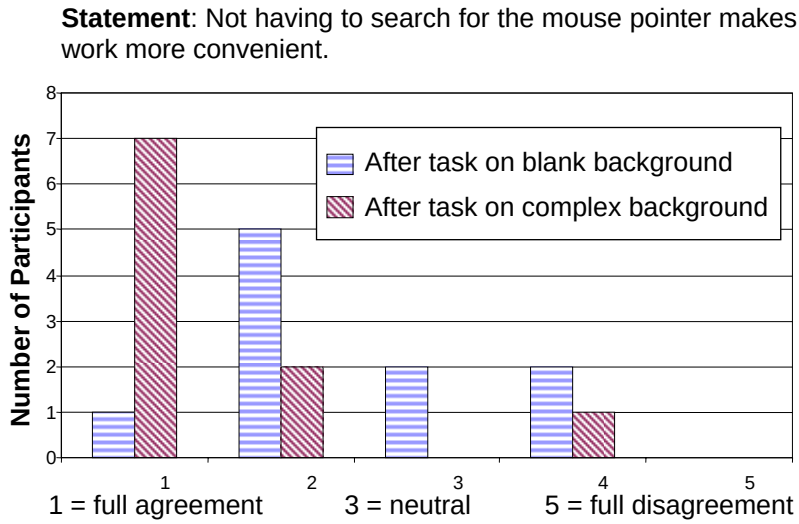

Fig. 11. After the task on the complex background the participants agreed even more that it is convenient not having to search for the mouse pointer.

5 Conclusions

The use of a gaze pointing technique can save a big portion of mouse movements without incurring additional load to the eye. The use of eye gaze pointing will therefore not fatigue the eye muscles more than with traditional mouse pointing. With the common trend to bigger displays, the distances covered by the mouse will become bigger. Gaze positioning in general seems to be a desirable input technology because it reduces the necessary mouse movements and less space is required for the mouse.

Gaze positioning with a touch sensitive mouse can bring a speed benefit compared to the classical mouse, especially when the user is not aware of the pointer position on the screen or when the distance to the target is very far. The reason why the touch sensitive mouse key does not produce the same speed as the gaze key introduced by

Ware and Mikaelian [15] lies in the time for checking whether the pointer is in the correct position after the touch. The gaze key triggers an action immediately without feedback on the gaze position. Gaze positioning with a touch-sensitive mouse gives feedback and the possibility to correct the position to achieve the accuracy demanded by the sizes of the GUI elements.

Combining the MAGIC pointing with a touch-sensitive mouse brings several advantages. The MAGIC pointing positions the mouse pointer at a first mouse movement and therefore needs compensation techniques to compensate the movement of the mouse. The MAGIC touch does not need such compensation, as the mouse does not move at the moment of gaze positioning. The concept of a first mouse move in MAGIC pointing implies a timeout and it is not clear how long this time should be. A user has to know whether the timeout expired already to predict the behavior of the system. The concept of touching a key makes the system easier to understand for the user. It also gives more control to the user because it offers also the possibility to reposition the mouse pointer immediately by lifting the finger and touching the key again.

Using the gaze as the only input modality has many severe challenges, but the combination with a touch sensitive mouse key solves some of them – the Midas Touch problem, the feedback issue and issues related to inaccuracy. The combination of an eye-tracker with a touch sensitive mouse key offers a way to utilize the gaze as an additional pointing device without changes in the behavior of the GUI.

There is only a little speed benefit in the case that the user is not aware of the mouse pointer position, but the savings in distance covered by the mouse are enormous. During the interviews, we got very positive feedback for the gaze positioning. The possibility to position the mouse pointer at the position of the gaze just with a little movement of the finger tip feels really pleasant. As a consequence of all, the use of gaze pointing is desirable.

Acknowledgements

The work has been conducted in the context of the research project Embedded Interaction (‘Eingebettete Interaktion’) and was partly funded by the DFG (‘Deutsche Forschungsgemeinschaft’).

References

1. Ashdown, M., Oka, K., and Sato, Y. Combining head tracking and mouse input for a GUI on multiple monitors. In *Extended Abstracts on Human Factors in Computing Systems*. CHI '05. ACM Press (2005), 1188-1191.
2. Bolt, R. A. Gaze-orchestrated dynamic windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and interactive Techniques*. SIGGRAPH '81. ACM Press (1981), 109-119.

3. Card, S. K., English, W. K. and Burr, B. J. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, *21*, (1978) 601-613.
4. Duchowski, A. Eye Tracking Methodology: Theory & Practice. Berlin: Springer-Verlag, 2003
5. ERICA eye tracker product description http://www.eyeresponse.com
6. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology*, (1954) 47, 381-391.
7. Hinckley, K. and Sinclair, M. 1999. Touch-sensing input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. ACM Press (1999), 223-230.
8. Jacob, R. J. The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. In *ACM Trans. Inf. Syst.*, (1991) 9(2), 152-169.
9. Kumar, M., Paepcke, A. and Winograd, T. EyePoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '07. ACM Press (2007), 421-430.
10. Majaranta, P. and Räihä, K. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. ACM Press (2002), 15-22.
11. Majaranta, P., Aula, A. and Räihä, K. Effects of feedback on eye typing with a short dwell time. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications.* ETRA '04. ACM Press (2004), 139-146.
12. Miniotas, D., Špakov, O. and MacKenzie, I. S. Eye gaze interaction with expanding targets. In *Extended Abstracts on Human Factors in Computing Systems.* CHI '04. ACM Press (2004), 1255-1258.
13. Qprox capacitive sensor QT 113 product description http://www.qprox.com/
14. Salvucci, D. D. and Anderson, J. R. Intelligent gaze-added interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '00. ACM Press (2000), 273-280.
15. Ware, C. and Mikaelian, H. H. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the CHI + GI '87*, ACM Press (1987), 183-188.
16. Zhai, S., Morimoto, C. and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '99. ACM Press (1999), 246-253
17. Zhang, Q., Imamiya, A., Go, K., and Mao, X. Resolving ambiguities of a gaze and speech interface. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 85-92