Eye-Gaze With Predictive Link Following Improves Accessibility as a Mouse Pointing Interface

Jason Vazquez-Li, Lyle Pierson Stachecki, John Magee
Department of Mathematics and Computer Science
Clark University, Worcester, MA, USA
{jvazquezli, lpiersonstachecki, jmagee}@clarku.edu

ABSTRACT

We propose a target-aware pointing approach to address one predominant problem in using eye-controlled mouse replacement software: the lack of high-precision movement. Our approach is based on Predictive Link Following [4], which alleviates the difficulties with link selection when using mouse replacement interfaces by predicting which link should be clicked based on the proximity of the cursor to the link. For cursor control via eye movement, an eye tracking algorithm was implemented using the Tobii EyeX device to detect and translate gaze location to screen coordinates. We conducted an experement comparing eye-gaze controlled mouse pointing with and without the Predictive Link following approach. Our results demonstrate increased accuracy of our system compared to just using eye-controlled mouse pointing.

Keywords

Accessibilty, Mouse Replacement Interfaces, Eye-gaze Interfaces, Target-Aware Pointing

1. INTRODUCTION

Eye-gaze controlled software is a common interface modality for people with severe motion disabilities. However, tasks such as mouse control are difficult due to the natural jittery movements of the eyes, which prevents fine-tuned movement of the mouse from being made [2]. As most software is not designed for eye-gaze input, having small interactive elements,users are limited in what interactions they can reasonably do, which can cause frustration.

Most eye-gaze systems do not provide for high-precision movements to control a mouse. We base our system on an approach originally designed for the Camera Mouse. The Camera Mouse allows users to control the mouse using certain body parts (e.g. tip of nose), but it does not allow eye movement as a user input option [1]. This is a problem for people who cannot move their heads. Given that, our system makes progress towards addressing the lack of high-precision movement when using eye-controlled mouse replacement software. We also base our approach on HMAGIC [3], a system that combines eye-gaze with the Camera Mouse. Other approaches include force fields, speed reduction, and warping [5]. Our approach here accumulates a score based on pointing *near a target* to assist in selecting the target even when the user is unable to actually click on it.

ASSETS '16 October 23-26, 2016, Reno, NV, USA

ACM ISBN 978-1-4503-4124-0/16/10.

DOI: http://dx.doi.org/10.1145/2982142.2982208

2. SYSTEM OVERVIEW

The Tobii EyeX eye tracking hardware was used for eye-gaze input. Traditional infrared eye tracking hardware has been cost-prohibitive, but the EyeX is a consumer-grade device, costing less than $200. Our system is arranged as follows:

2.1 Controlling Cursor Movement

The X and Y coordinates of the user's eye relative to the monitor are brought into our system and used to calculate the screen coordinate of where the mouse should be positioned. As a pre-processing step, in order to reduce the amount of mouse jitter, we set the mouse position to the average position of the last 20 gaze points. A gaze point is registered about every 15 milliseconds.

2.2 Clicking

Clicks are triggered by blinking of the eyes; a blink lasting approximately 275 milliseconds will trigger a click. This works because the EyeX device stops detecting gaze movement if the user closes their eyes. Thus, our system checks for interruption in gaze movement for a specific span of time to perform the clicks. To actually click, a "leftClick" method implements the system functions that simulate pressing and releasing the left mouse button.

2.3 Modification of Predictive Link

We modified the the Predictive Link function to better work for eye-gaze input. The original algorithm [4] had a discontinuity as the distance from the pointer to a target approached zero:

$$\text{pointer}(a) = \begin{cases} \frac{1}{\text{dist}(a, clickpoint)^{\beta}} * \gamma & \text{if clicked on page} \\ hoverScore & \text{otherwise,} \end{cases}$$

where hoverScore is the value assigned when the pointer is over a link.

In our system, a slight modification provides a function that approaches a constant as the pointer gets close to a link:

$$\text{pointer}(a) = \frac{1}{(\text{dist}(a, mousepoint) + 1)^{\beta}} * \gamma$$

The function is used to accumulate a score for how likely a user is trying to click on a particular link a based on the distance from the pointer to the link. β and γ are configurable parameters.

3. EXPERIMENTAL EVALUATION

A preliminary experiment was conducted to evaluate our software.

3.1 Apparatus, Procedure, and Users

Two versions of a website containing various-sized hyperlinks were used to test our system. The first version of the website (condition A) included the necessary JavaScript files for the Predictive Link technology software; the second version (condition B) did not. This was done to compare the effectiveness of the two different methods. Another script file running in the background for both website versions randomly selects and highlights a hyperlink on the website for the user to click on. The target link will remain highlighted for 5000 milliseconds before the script selects another hyperlink, with a pause time of 1500 milliseconds between trials.

For the website with the Predictive Link function, the user does not have to click directly on the link to register it as a successful click, rather, the output of the prediction algorithm is used to determine if the link was clicked. In contrast, the user must directly click on the hyperlink if using the website without the Predictive Link function. For each trial, the evaluation software stores either a successful click or a failed click (not clicking on the assigned hyperlink within the time limit, or clicking on another link). Each participant conducted 50 trials for each condition. At the end of the test session, the result is reported as a percentage.

Seven people without motion disabilities participated in our test. Each participant was given the same instructions and assigned a random order of conditions. That is, those that started with condition B had the first test as the webpage without the Predictive Link function, and then completed condition A, with the second webpage with the Predictive Link function. Six of the participants had minimal experience with eye gaze input. User 7 had been using eye gaze input for several weeks. In total, there were 700 trials conducted (7 participants, 2 conditions, 50 trials each).

3.2 Results and Discussion

Table 1 summarizes our results. Excluding User 7 (because that participant had significantly more experience with eye gaze) the means, standard deviations, and 90% confidence interval for a means difference test are as follows.
Condition A (with Predictive Link) mean: 35.7%
Condition B (without Predictive Link) mean: 21.7%
Condition A standard deviation: 13.8
Condition B standard deviation: 10.6

The 90% confidence interval of the means, calculated using the Student's t-distribution is [1.1, 26.9].

Table 1: Results

User	With Predictive Links	Without Predictive Links	First Condition
1	46% Correct	26% Correct	A
2	58% Correct	40% Correct	A
3	26% Correct	10% Correct	B
4	34% Correct	22% Correct	B
5	22% Correct	14% Correct	A
6	28% Correct	18% Correct	B
7	94% Correct	56% Correct	A

Subjects three and five had problems with the calibration of the Tobii EyeX.

Based on the results, using Predictive Link technology increased performance by 14%. However, even though using Predictive Link boosted performance significantly, the mean success rate of 35.7% for Condition A is still not high enough for a reliable interaction system. A contributing factor may be lack of experience using Tobii EyeX. We can see this in User 7, who scored 94% on the Condition A test (with Predictive Link). This participant was one of the developers of this software and thus used the device extensively; further testing will evaluate whether there is a learning effect on performance.

4. CONCLUSION

Pairing eye-controlled mouse replacement software with Predictive Link technology increased accuracy for tasks demanding high-precision cursor movement. Although the performance of inexperienced users in our initial test was not high, it seems likely that accuracy increases for users who have been using the system for longer periods. Future studies with more participants will examine this as well as provide data for further analysis of statistical significance.

5. ACKNOWLEDGMENTS

The authors would like to thank Jiri Roznovjak for his asstance in modifying the predictive link algorithm.

6. REFERENCES

[1] M. Betke, J. Gips, and P. Fleming. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. IEEE Trans. Neural Systems and Rehabilitation Engineering, 10:1-10, 2002

[2] R. J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. ACM Trans. Inf. Sys., 9(2):152-169. Apr. 1991

[3] A. Kurauchi, W. Feng, C. Morimoto, and Margrit Betke. HMAGIC: head movement and gaze input cascaded pointing. PETRA '15, 4 pages. 2015.

[4] J. Roznovjak, and J. Magee. Predictive Link Following for Accessible Web Browsing. ASSETS'15, pp 403-404. Oct. 2015.

[5] X. Zhang, X. Ren, and H. Zha. Improving eye cursor's stability for eye pointing tasks. CHI '08, pp 525-534. 2008.