eyeCan: Affordable and Versatile Gaze Interaction

Sang-won Leigh
MIT Media Lab
75 Amherst St, Cambridge, MA 02139
sangwon@mit.edu

ABSTRACT

We present *eyeCan*, a software system that promises rich, sophisticated, and still usable gaze interactions with low-cost gaze tracking setups. The creation of this practical system was to drastically lower the hurdle of gaze interaction by presenting easy-to-use gaze gestures, and by reducing the cost-of-entry with the utilization of low precision gaze trackers. Our system effectively compensates for the noise from tracking sensors and involuntary eye movements, boosting both the precision and speed in cursor control. Also the possible variety of gaze gestures was explored and defined. By combining eyelid actions and gaze direction cues, our system provides rich set of gaze events and therefore enables the use of sophisticated applications e.g. playing video games or navigating street view.

Author Keywords

Gaze interaction; gaze tracking; accessibility

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Input devices and strategies

INTRODUCTION

Gaze interaction plays a crucial role in improving the quality of life of people with severe motor disability. But the problem is that, though there are high-quality gaze trackers off the shelf (e.g. Tobii, QuickGlance), such devices are not easily affordable due to high price and small market size. Furthermore, despite the rich amount of academic and industrial efforts that has been spent, gaze interaction is not enough usable for beginner level users and therefore requires intensive user education. Mainly it is because human eyes were more responsible for subtle communications or perceiving information than command-and-control [1]. So these problems of high-cost and low-usability are summed up to make gaze interaction systems impractical.

In this research, we tackle this problem by providing a usable gaze interaction system *eyeCan* which is built upon the utilization of low-cost DIY eye-trackers. A big problem we had with this low-cost setup is that the hardware configuration does not promise stability and accuracy for mouse cursor control. In order to make such low-precision trackers usable, we first introduce a simple but powerful motion filtering method to reduce the effect of noise and inaccuracy in gaze tracking. The proposed filter let users control cursors accurately without much compromising the responsiveness and quickness.

From aspect of usability, the major limitation of currently available gaze interaction systems is that they are mainly concerned with point-and-select. Because beginner level users don't have enough dexterity with eye controls, only relying on such point-and-select interaction, it is nearly impossible for the users to work with sophisticated applications (that requires bunch of key or button actions). They should click the buttons on onscreen keyboard one by one or stressfully navigate through long menu trees back and forth. There were previous efforts to alleviate such difficulties by providing richer gaze interactions - providing full gaze control in multiple online gaming environments [2], or defining a structured set of gaze events [3]. But the proposed interactions were not fully generalized or not comprehensive, and then, we found out that the set of generalized gaze events can be largely extended through combining eyelid actions with gaze direction cues. In this paper, we present an extended set of gaze events and their applications along with a practical framework for defining customized, app-specific gaze interactions.

Figure 1: $50 gaze tracker used for our experiment

SYSTEM DESIGN

As in Fig 1 our experimental hardware consists of an IR camera and IR LED array, and it captures eye movement at speed of 60 frames per second. Then the computed triplet of data X, Y, b (X, Y: 2-D coordinate of gaze point, b: eyelid open/close) is transferred to *eyeCan* system.

(1) Improving Gaze-pointing via Adaptive Motion Filter

Our system adaptively filters the gaze input (X, Y) to compute the actual motion of a cursor – the cursor moves and adapts slowly (with low sensitivity) to the gaze position

UIST'13 Adjunct, October 8–11, 2013, St. Andrews, United Kingdom.
ACM 978-1-4503-2406-9/13/10.
http://dx.doi.org/10.1145/2508468.2514719

when the eye movement is small, and adapts quickly (with high sensitivity) when the eye movement is large and fast. This technique effectively reduces the noise from gaze trackers and let cursors move precisely without much jittering, also keeping the cursor speed high when eyes move fast. When gaze tracking data contains noise of standard deviation of ~150 pixels (7cm in our setup), users could successfully make the cursor stay stationary in a box size of 1.5cm x 1.5cm without much visual fatigue.

Another crucial benefit is that users can look around the cursor without shifting its position (thanks to low sensitivity when eye movement is small), so that they can incorporate surrounding views to precisely and correctly locate the cursor. This also effectively alleviates the Midas touch problem [5] by temporally differentiating the gaze pointing action from visual perception.

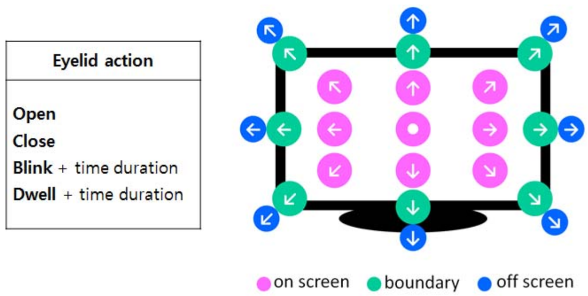

Fig 2: Eyelid actions and gaze positions. 4x25=100 or more combinations are available

(2) Providing Rich Set of Gaze Events

Eyelid gestures are spatially invariant: eye open/close, blinking, and dwelling (keeping open). The number of given spatially invariant gestures is very small, but a much richer modality is possible if such gestures are combined with gaze direction (spatial) information. As in Fig 2, points on and off the computer screen can be divided into groups that can be used for discriminating gaze events according to its point of trigger. A similar approach had been done by Kumar and Winograd who proposed scrolling techniques using gaze direction cues [4], but our system aims to apply gaze control to a much broader applications through extended variation with eyelid actions. One representative example would be flipping pages of an e-book by just looking at the corner of the screen, or a much complex interaction such as playing video games can be provided by assigning directional pad to gaze direction and button input to eye blinks.

Especially users with limited eye motion capability can greatly benefit from this combined event set, because a much easier gaze interactions than point-and-select is possible. For example, a severely paralyzed user may not want to do a full control over windows OS, but only wants to flip the page of an e-book application. Similarly, there will be a number of different app-specific needs, so it seems very important to support such varying desires. *Action-Script*, a simple XML-style grammar, allows users to define app-specific interactions by mapping gaze events to OS events. It also provides much complicated variations like programming parallel or sequential gaze controls. For example, by assigning two or more different tasks to a single gaze event, users can perform a sequence or combination of tasks with a single eye blink or any other gesture - allowing virtually infinite variations.

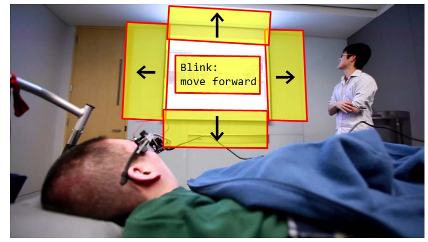

Figure 3: Gaze Interaction for Street View navigation

RESULT & FURTHER WORK

We have tested our system with users whose bodies are completely paralyzed. Users who have previous experience with commercial eye trackers had no problem at using this device for a casual computer use, and entry-level users were able to control on-screen keyboard after a few hours of training (1 hours a week, for a month). From the tests, *eyeCan* system seems to work greatly even with low-precision gaze trackers in aspect of usability, but for a more practically applicable product, we conclude that additional factors such as context-awareness and user pattern analysis will be required to boost the user 's comfort and speed of interaction.

ACKNOWLEDGMENTS

This project was supported by Samsung Electronics, as the first project of Samsung's Creative Lab.

REFERENCES

1. Hyrskykari, A., Majaranta, P. and Räihä, K-J. From Gaze Control to Attentive Interfaces. *In Proc. HCII 2005*
2. Istance, H., Vickers, S., Hyrskykar, A., Gaze-Based Interaction with Massively Multiplayer On-line Games. *Ext. Abstracts CHI 2009*. ACM Press (2009), 4381-4386
3. Kangas, J. et al. Defining Gaze Interaction Events. *"Gaze Interaction in the Post-WIMP World" Workshop of CHI'13*
4. Kumar, M., Winograd, T. Gaze-enhanced Scrolling Techniques. In Proc. *UIST '07.* ACM Press (2007)
5. Velichkovsky, B., Sprenger, A., and Unema, P. Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem". In *Proc. the IFIP TC13 International Conference on Human-Computer Interaction* (1997), 509-516